(1)

A ROBUSTNESS ANALYSIS OF MOVING-BANK MULTIPLE MODEL ADAPTIVE ESTIMATION AND CONTROL OF A LARGE FLEXIBLE SPACE STRUCTURE

THESIS

Daniel F. Van Der Werken, Jr.
Captain, USAF

AFIT/GE/ENG/88D-59

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

(1)

AFIT/GE/ENG/88D-59

A ROBUSTNESS ANALYSIS OF MOVING-BANK MULTIPLE MODEL ADAPTIVE ESTIMATION AND CONTROL OF A LARGE FLEXIBLE SPACE STRUCTURE

THESIS

Daniel F. Van Der Werken, Jr.
Captain, USAF

AFIT/GE/ENG/88D-59

AFIT/GE/ENG/88D-59

A ROBUSTNESS ANALYSIS OF MOVING-BANK MULTIPLE MODEL ADAPTIVE ESTIMATION AND CONTROL OF A LARGE FLEXIBLE SPACE STRUCTURE

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Electrical Engineering

Daniel F. Van Der Werken, Jr., B.S.
Captain, USAF

December 1988

Approved for public release; distribution unlimited

Preface

The purpose of this thesis is to determine the effects of unmodeled states on moving-bank multiple model adaptive estimation and control of a large flexible space structure. Previous research has shown that moving-bank multiple model adaptive estimation and control is effective when the order of the filter model is equal to the order of the truth model. This research determines whether moving-bank multiple model adaptive estimation and control is a viable technique when one (inevitably) uses a reduced order filter model, or if the unmodeled effects seriously degrade or confound the adaptation process. Emphasis is placed on state estimation rather than parameter estimation. The results of this thesis research show that the performance of the moving-bank algorithm for state estimation and control is seriously degraded, and the ability of the algorithm to produce accurate parameter estimates is totally confounded by the unmodeled states.

I thank Dr. Peter S. Maybeck for his guidance, patience, continual enthusiasm, and all important encouragement throughout my thesis effort. I give great thanks to my wife, [redacted] for being loving, patient, and extremely supportive during my AFIT assignment. I also thank my [redacted] daughter, [redacted] for being one of the joys of my life, and finally I give thanks to Jesus Christ, my Lord and Savior, for giving me the abilities to perform this work and handle the pressures involved.

Daniel F. Van Der Werken, Jr.

Table of Contents

List of Figures

List of Tables

AFIT/GE/ENG/88D-59

Abstract

This thesis study is a robustness analysis of a moving-bank multiple model adaptive estimation and control algorithm. The truth model consists of 24 states, and the filter model consists of 6 states. The object is to determine whether or not the state mismatch between the filter and truth models seriously degrades or totally confounds the adaptation process.

A rotating two-bay flexible space structure, approximating a space structure that has a hub with extending appendages, is the system model used in this research. The mass of the hub is greater than the mass of the appendage, and finite element analysis is used to develop the mathematical model. The system is developed in physical coordinates, transformed to modal coordinates, and the method of singular perturbations is used to provide a reduced filter model. The two dimensional parameter space consists of variations of the mass and stiffness matrices for the two-bay truss.

Results indicate stable and accurate state estimation when the bank is initially centered on the true parameter. Stable control and accurate parameter estimation are not achieved. When the bank is initially centered at an incorrect parameter, stable and accurate state estimation is not demonstrated. These results indicate a total confounding of the adaptation process due to the unmodeled states, since accurate estimation and control can be generated when there are no such unmodeled states.

MOVING-BANK MULTIPLE MODEL ADAPTIVE ESTIMATION APPLIED TO FLEXIBLE SPACE STRUCTURE CONTROL

1. Introduction

One of the areas of stochastic estimation that is currently undergoing research is the area of Multiple Model Adaptive Estimation (MMAE) and Control (MMAC) [1; 2; 4; 5; 8; 9; 10; 14; 15; 16]. Uncertain parameters in mathematical system models cause problems in some estimation and control applications. These problems occur when uncertain parameters cause significantly reduced accuracy in the model and algorithms based upon that model. These uncertain parameters can have values which are unknown constants, slowly changing variables, or abruptly changing variables. Adaptive estimation and control is performed when the parameters within the mathematical models are identified in real time based upon information about the values of the changing system parameters of an estimation or control problem. Multiple Model Adaptive Estimation and Control is one important approach to this type of adaptive control problem.

The central issue of this thesis research is robustness. Previous research demonstrated that, if measurements are sufficiently accurate and if a 6-state model of a flexible space structure is adequate, then moving-bank multiple model adaptive estimation and control can be very effective [8:196-197]. This

research will attempt to determine the effects of the order mismatch between a 24-state truth model for such a flexible space structure and a 6-state filter model, assuming that a 24-state linear model, in fact, represents the real world adequately. The real purpose is to see if moving-bank multiple model adaptation is a viable technique when one (inevitably) uses reduced order models, or if the unmodeled effects seriously degrade or totally confound the adaptation process. During this thesis research, emphasis will be placed on the ability to estimate and control the states rather than the ability to estimate parameters: parameter estimation accuracy is important only to the degree that it enhances state estimation and control.

1.1 Background

Basically, MMAE and MMAC are methods of providing estimation and control for a system with uncertain parameters. The estimation and control is performed by using a bank of elemental Kalman filters. A Kalman filter can be defined according to the following [13:4]: "A Kalman filter is simply an optimal recursive data processing algorithm." Figures 1 and 2 are diagrams which describe how MMAE and MMAC are performed. Each Kalman filter is based upon an assumed parameter value selected from the set, $\underline{a}_1, \underline{a}_2, \ldots, \underline{a}_K$. Based on these assumed parameter values, each Kalman filter generates state estimates, $\hat{\underline{x}}_1, \hat{\underline{x}}_2, \ldots, \hat{\underline{x}}_K$, and residuals, $\underline{r}_1, \underline{r}_2, \ldots \underline{r}_K$, when the measurement $\underline{z}_i$ becomes available at time t_i. The residuals are then processed

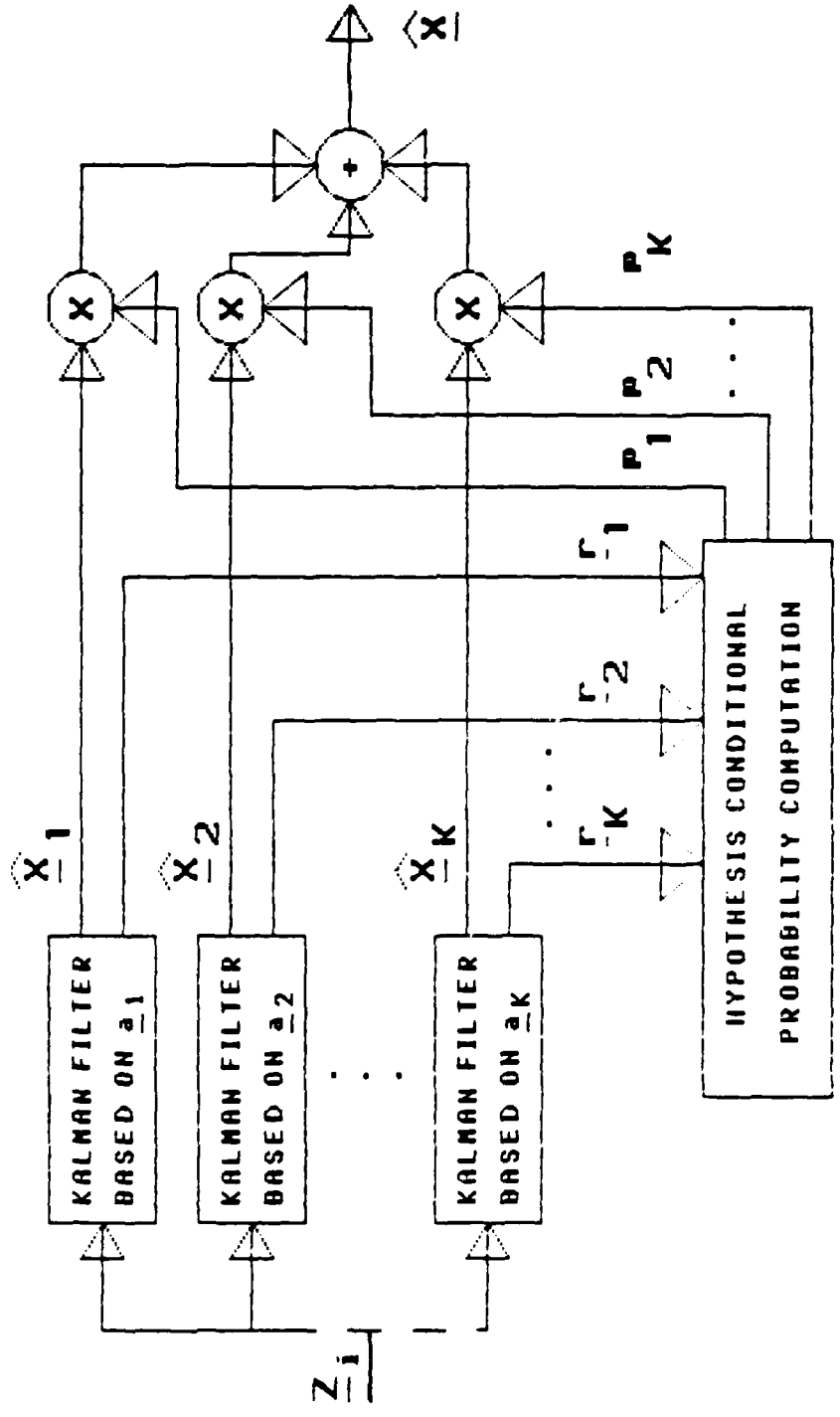

Figure 1

Moving-Bank Multiple Model Adaptive Estimation

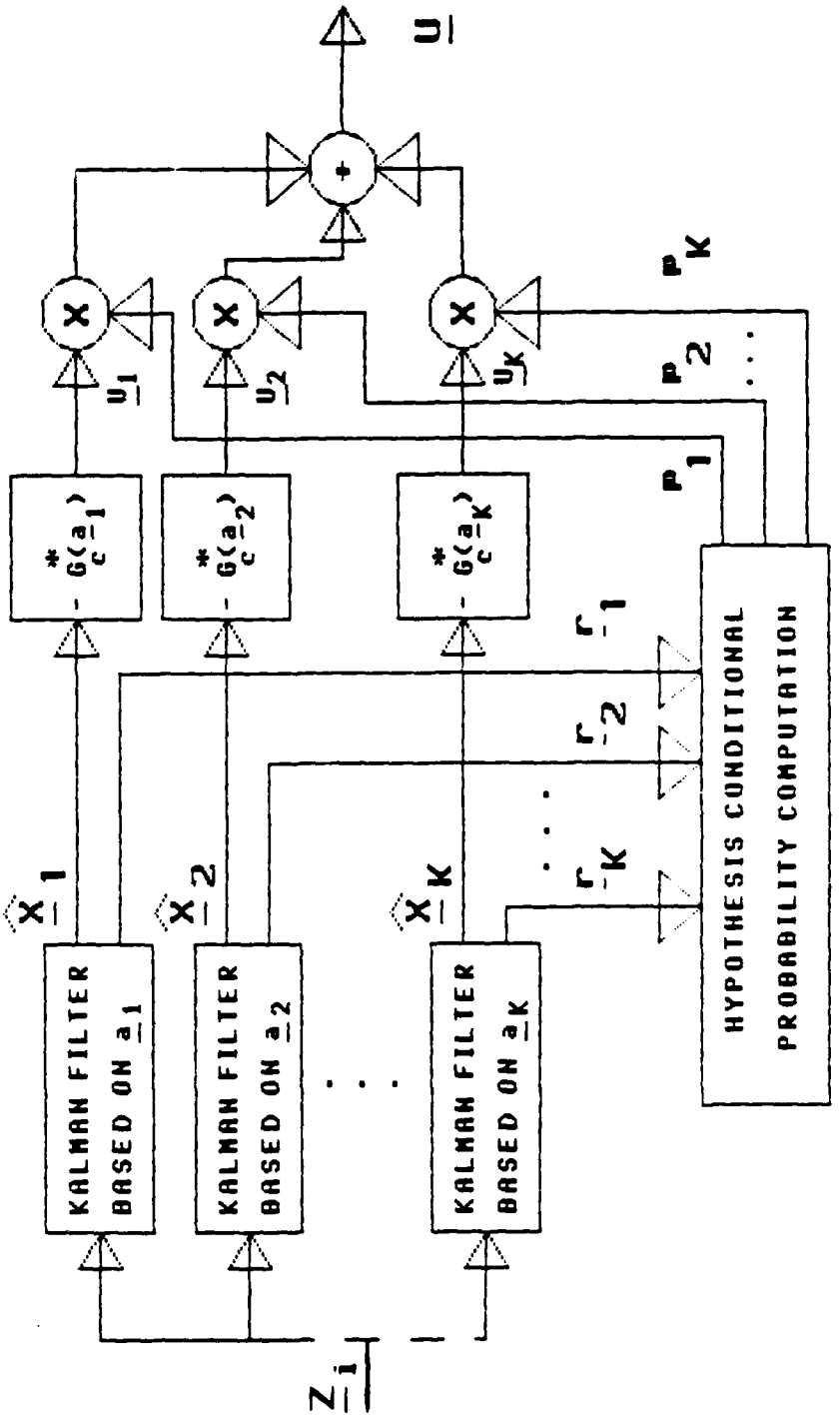

Figure 2

Moving-Bank Multiple Model Adaptive Control

by a conditional probability computation method to weigh the outputs of those elemental filters to provide the best estimate of states of the system as a probabilistically weighted sum. [14:120-132]

The MMAC algorithm works in a similar manner to the MMAE algorithm except that the state estimates are provided as inputs to optimal controller gains, $-G_c^*(\underline{a}_1)$, $-G_c^*(\underline{a}_2)$,..., $-G_c^*(\underline{a}_K)$, to generate the control inputs, $\underline{u}_1$, $\underline{u}_2$,..., $\underline{u}_K$. The control inputs are probabilistically weighted and added together to provide the adaptive control. [15:253-255]

1.1.1 Fixed and Moving-Bank Configurations

There are basically two types of Multiple Model Adaptive Estimators and Controllers: fixed-bank and moving-bank. Figures 3, 4, 5, and 6 describe the fixed and moving-bank configurations. The fixed-bank MMAE relies on a stationary bank of Kalman filters configured in such a way (refer to Figure 1) as to estimate the changing system parameters, and the moving-bank MMAE relies on a moving bank of Kalman filters configured in a like manner to do the same. As shown by Figures 3, 4, 5, and 6, the parameter space is assumed to be two dimensional with each dimension having 10 discrete values.

Figure 3

Fixed-Bank MMAE/MMAC -- Fine Discretization

Figure 4

Fixed-Bank MMAE/MMAC -- Coarse Discretization

Figure 5

Moving-Bank MMAE/MMAC -- Bank Move

Figure 6

Moving-Bank MMAE/MMAC -- Bank Expansion

1.1.2 Fixed-Bank Algorithm

Using a fixed-bank MMAE algorithm, the problem of uncertain parameters can be solved for a system which can be adequately represented by a linear stochastic state model. This MMAE algorithm takes the structure of Figure 1. For this solution, a Kalman filter must be designed for each possible parameter value in the parameter space, creating a bank of K separate Kalman filters.

For example, there are 100 separate Kalman filters in the fixed-bank MMAE shown in Figure 3. These Kalman filters are a result of having a two dimensional parameter space with 10 discrete values for each scalar parameter. As shown in Figure 3, an elemental Kalman filter may not necessarily be located at the exact position as the 'true' parameter value. In this case, the elemental filters nearest the 'true' parameter value work together to produce the current best estimate of the states at this 'true' parameter value.

In general, the parameter space is continuous, resulting in an infinite number of Kalman filters, but a K-point discrete approximation is used. Residuals for each Kalman filter are calculated when measurements become available. These residuals have the property such that: "...one would expect that the residuals of the Kalman filter based upon the 'correct' model will be consistently smaller than the residuals of the other

mismatched filters." [14:133]. The conditional probability produced by the bottom block of Figure 1 for the "correct" elemental Kalman filter will be larger than the conditional probability values for the other elemental Kalman filters. Using these conditional probability values as weighting factors, the adaptive state estimate is produced. This adaptive estimate is determined by one of two means: probabilistically weighted averaging of each Kalman filter output (called Bayesian MMAE estimation) or allowing the overall state estimate be equal to the state estimate of the filter with the highest conditional probability density value (called maximum a posteriori, or MAP, estimation). [14:129-136; 8:2-4; 16]

The large number of elemental Kalman filters required for fixed-bank MMAE causes a problem. For the example given in Figure 3 with two uncertain but constant parameters, each of which can assume 10 possible discrete values, the fixed-bank MMAE require $10^2 = 100$ separate filters. Given more uncertain parameters with a higher number of discrete values, the fixed-bank MMAE implementation becomes computationaly intensive. When the parameter temporal variations are modeled as a Markov process or by other means, the number of elemental Kalman filters required for fixed-bank MMAE is prohibitive. [14:129-136; 4:2]

1.1.3 Moving-Bank Algorithm

The moving-bank MMAE implementation relieves this problem by only requiring nine separate filters for the same example described previously (see Figure 5). The moving-bank MMAE creates a movable bank of elemental filters "...associated with the parameter values that most closely surround the estimated 'true' parameter value" [8:7].

The moving-bank MMAE behaves the same as the fixed-bank MMAE when the true parameter value lies within the region covered by the nine movable filters. When the true parameter lies outside or on the boundary of this region, the moving-bank MMAE must either move (see Figure 5) or expand (see Figure 6) the bank of filters to accommodate this new value [2:27-28; 8:7]. Research conducted during the past four years has shown that moving-bank MMAE algorithms can significantly reduce the amount of computational loading required to produce estimates, while preserving the essential adaptive estimation attributes of a full fixed-bank MMAE algorithm [1; 2; 4; 5; 8; 9; 14; 16].

1.2 Past Research

Hentz [2; 16], who conducted the initial research, compared the moving-bank and fixed-bank MMAE algorithms against each other on a simple two state system model. Hentz found that "...the sliding bank estimator/controller offers performance that is essentially identical to that of a single Kalman filter/LQ

controller that has (artificial) knowledge of the true parameters, once parameter acquisition has occurred" [2:108]. Hentz also found that, for a limited computational level, a cruder discretization was forced upon the fixed-bank MMAE, causing the performance of the moving-bank MMAE to increase over that of the implementable fixed-bank MMAE as the original parameter space discretization became finer or the "...number of uncertain parameters increased" [2:108].

Filios [1], who conducted follow-on research on a more complex model, discovered that ambiguity function analysis [14:96-101] is useful for moving-bank MMAE design because it provides information about the performance of a parameter estimator. The ambiguity function is basically the average value, over all possible measurement histories, of the likelihood function for the given problem. Its shape reveals potential ambiguity problems (as exhibited by multiple peaks) and parameter estimation accuracy (as shown by its curvature at the global peak). However, Filios also found that his more complex system model did not really require adaptive control. [1:93]

Following Filios, Karnick [4; 5] applied the moving-bank algorithm to a different but still complex system. This system is the 13-member two-bay truss developed by Karnick in his thesis [4:45-58] and is pursued further in this research. For this system, Karnick found that adaptive estimation and control was

required. However, Karnick's MMAE was unable to identify the truth model parameter accurately, even though it could produce accurate estimates of the states. By making sure the truth model parameter was contained within the parameter space points of the fixed-bank MMAE, Karnick was able to show that the fixed-bank MMAE performed just as well as or better than the moving-bank MMAE. Karnick also found that, instead of converging to a consistent parameter value, the bank of the moving-bank MMAE wandered about the parameter space. [4:5,78,93]

Lashlee [8; 9], following Karnick, directed his research toward finding out why the moving-bank MMAE did not perform as well as the fixed-bank MMAE for Karnick's more complex system model. Focusing his attention on the same system model that Karnick used, Lashlee found that having the noise strength matrices (dynamics noise strength Q and measurement noise covariance R) too large masked the distinctions between good and bad models. Reducing the real world noise values yielded better results. With improperly valued Q and R matrices, the moving-bank MMAE wandered about the parameter space and performed less well than the fixed-bank MMAE. Once the Q and R matrices were set at sufficiently low values, Lashlee showed that the moving-bank MMAE was able to find the parameter value and outperform the fixed-bank MMAE. [8:9,198]

1.3 Problem

Even though moving-bank MMAE has been shown to require less computational loading than fixed-bank MMAE, the problem remains of investigating the robustness of the moving-bank MMAE. Investigating the robustness issue involves determining the effects of the order mismatch between a 24-state truth model and a 6-state filter model. The 24-state linear truth model is assumed to represent the real world adequately. This thesis investigation will concentrate on discovering whether or not the use of moving-bank MMAE is a viable technique when a reduced order model is inevitably used and how seriously the adaptation process is degraded (or even totally confounded) by the unmodeled effects.

1.4 Scope

This thesis research concentrates in the area of moving-bank Multiple Model Adaptive Estimation (MMAE) and Control (MMAC). In particular, this effort will concentrate on the design of controllers for a flexible space structure. This space structure is the two-bay truss developed by Drew Karnick in his thesis [4:45-58]. Two cases for the model for this structure are considered. The first case models the truss as a 6-state system using one rigid body mode and two bending modes. The second case models the truss as a 24-state system having one rigid body mode and 11 bending modes. For each case, the state estimates produced are a position and velocity estimate for each mode.

This thesis effort is a continuation of the studies performed by Hentz, Filios, Karnick, and Lashlee. In particular, this thesis effort will focus on the recommendations made by Lashlee in his thesis. Lashlee's thesis found these conclusions:

1. The values chosen for the R matrix (measurement noise covariance matrix) determine whether or not the moving-bank algorithm performs correctly. When the R matrix values were too large, the distinction between good and poor models was masked. This masking caused the moving bank of filters to wander aimlessly without providing useful estimates of the states. Lowering the R matrix values yielded better results.

2. The state weighting matrix and the control weighting matrix of the linear Gaussian quadratic (LQG) regulator synthesis should be tuned for each parameter value used by the moving-bank MMAC. The $G_c^*(\underline{a}_k)$ values shown in Figure 2 are based on choices of the weighting matrices in the quadratic cost to be minimized. Having the state weighting matrix and the control weighting matrices of the LQG regulator properly tuned is important for optimal MMAC function. Lashlee determined the tuned state weighting matrix (X) values by:

> ...holding the control weighting matrix (U) constant and increasing the X values one at a time until the RMS values for the true states stop decreasing drastically as the X values were increased. [8:87]

The control weighting matrix (U) values were determined in a similar manner after the X values were found:

> ...the U values were found by holding the X values constant and decreasing the U values until the RMS values for the true states stop decreasing drastically as the U values were decreased. [8:87]

Note that the desire is not to provide fine tuning for the state and control weighting matrices, but the desire is rather to coarsely tune the matrices for the reasonable performance required for this type of design study. A final (operational) implementation of the algorithm would require a greater deal of state and control weighting matrix fine tuning.

3. When the true parameters did not correspond to one of the parameter values associated with any one of the elemental filters in the fixed-bank MMAE (which necessarily used coarser discretization), the moving-bank MMAC outperformed the fixed-bank MMAC. When the true parameters did correspond to one of the parameter values associated with one of the elemental filters in the fixed-bank MMAE, the performance of the moving-bank MMAC and the fixed-bank MMAC were about the same.

Based on his conclusions, Lashlee established a set of recommendations. These are:

1. Continue to use the two-bay truss model for the large flexible space structure.

2. Look into the impact of unmodeled effects (robustness issue) by analyzing the performance of using larger-than-6-state truth models and a 6-state filter, as compared to the 6-state truth model and the 6-state filter used in his thesis.

3. Tune the state weighting and control weighting matrices for each parameter point.

1.5 Approach

Based on Lashlee's recommendations, this research will consist of the following steps. First, it will continue to use the rotating two-bay truss model for the large flexible space structure. The structure remains the same as the one Lashlee used in his thesis, with 12 modes and 12-dimensional mass and stiffness matrices, resulting in a 24-state truth model. Using the method of singular perturbations [6:123-132; 7:29-37], the order of the system can be reduced to a 6-state truth model [4:52-58; 8:61-69]. Second, Lashlee's thesis used this 6-state truth model and a 6-state filter model; this thesis will focus on the effect of the order mismatch between the original 24-state truth model and the reduced-order 6-state filter model.

The central issue of this thesis research is robustness. Previous research demonstrated that, with an accurate 6-state model and accurate measurements, moving-bank multiple model adaptive estimation and control can be very effective. The effects of the order mismatch between the 24-state truth model and the 6-state filter will be the main area of investigation of this research. The underlying assumption is that the 24-state linear model does represent the real world adequately. Reduced order filter models are almost always used for online implementation. The intent of this research is to show whether or not moving-bank multiple model adaptation is a practical technique when the reduced order filter model is used. Of major concern is whether or not the unmodeled effects seriously degrade or totally confound the adaptation process. The following methodology is used.

1.6 Robustness Study

The robustness study is performed by changing the state dimensionality of the truth model to 24 states, but leaving the filter model unchanged as the 6-state model. In other words, the truth model and filter models are mismatched in dimensionality. This robustness study will allow the examination of the moving-bank filter performance as the truth model increases in complexity. The real issue is how well the adaptation process, and specifically the bank moving logic, performs in the face of phenomena in the real world that are not included in the filter's

reduced-order model. This study is the main crux of the thesis and will contain the bulk of the research. This study consists of three specific cases:

1. A single 6-state non-adaptive filter versus the 24-state truth model. This is the worst case. This case is used as a baseline against which the other cases is judged.

2. A single 6-state artificially informed non-adaptive filter versus the 24-state truth model. This is the best possible performance achievable from the implementable 6-state filter, without the effect of imperfectly estimated parameters. This case is used as another baseline for the best possible performance for a reduced-order state estimator.

3. The 6-state adaptive filter versus the 24-state truth model. This case is the true robustness test when the results are compared to those of Lashlee. It will determine how closely the filter can estimate the true values of states and how well the associated controller can regulate those states. For the sake of comparing to like quantities, the errors to be investigated are position and velocity errors at particular points on the truss, due to the combined effect of all modes (three in the filter model and 12 in the truth model). Parameter identification itself is only a secondary measure of performance.

The analysis on the three cases previously listed is performed using the moving-bank algorithm. The measure of performance used is the closeness of the filter state estimates to the true values and the quality of the state control. This performance analysis is conducted in the following manner:

1. Develop the computer algorithms for the 6-state non-adaptive, 6-state adaptive, and 6-state artificially informed moving-bank filters.

2. Run the filter and truth model configurations listed above.

3. Ensure the Q and R values of each configuration are set to give the minimum RMS error (tuning is an important issue with dimensionality mismatch between filter and truth models).

4. Evaluate the performance of the moving-bank filters by measuring how closely the filter estimates of positions and velocities at physical points on the truss match the true values, and then how well the associated controller regulates those states.

1.7 Summary

This chapter has covered the introduction, background, problem statement, scope of work, and approach taken. The

following chapters cover the algorithm development for the moving-bank MMAE and MMAC (Chapter 2), the development of the large space structure model (Chapter 3), a description of the simulations used in the thesis (Chapter 4), the results of the research and discussion (Chapter 5), and finally the conclusions, recommendations and discussion (Chapter 6).

II. Algorithm Development

2.1 Introduction

The algorithms for the fixed-bank and moving-bank Bayesian MMAE are developed in this order in this chapter: the fixed-bank first and then the moving-bank. Section 2.2 outlines the Bayesian multiple model adaptive estimation algorithm. The moving-bank algorithm is developed in Section 2.3, and Section 2.4 details the stochastic controller design.

2.2 Bayesian Multiple Model Adaptive Estimation Algorithm Development

The fixed-bank MMAE algorithms are formulated in this section. A more rigorous development is available in [14:129-136].

Considering a discrete-time linear time-invariant system [8:19]:

$$\underset{\sim}{x}(t_{i+1}) = \Phi(t_{i+1},t_i)\underset{\sim}{x}(t_i) + B_d(t_i)\underset{\sim}{u}(t_i) + G_d(t_i)\underset{\sim}{w}_d(t_i) \qquad (1)$$

$$\underset{\sim}{z}(t_i) = H(t_i)\underset{\sim}{x}(t_i) + \underset{\sim}{v}(t_i) \qquad (2)$$

where "$\underset{\sim}{\ }$" denotes a vector stochastic process and:

$\underset{\sim}{x}(t_i)$ = n-dimensional state vector

$\Phi(t_{i+1},t_i)$ = state transition matrix

$\underset{\sim}{u}(t_i)$ = r-dimensional known input vector

$B_d(t_i)$ = control input matrix

$\underset{\sim}{w}_d(t_i)$ = s-dimensional white Gaussian dynamics noise vector

$G_d(t_i)$ = noise input matrix

$\underset{\sim}{z}(t_i)$ = m-dimensional measurement vector

$H(t_i)$ = measurement matrix

$\underset{\sim}{v}(t_i)$ = m-dimensional white Gaussian measurement noise vector

The following statistics apply to Equations (1) and (2):

$$E[\underset{\sim}{w}_d(t_i)] = \underline{0} \tag{3}$$

$$E[\underset{\sim}{w}_d(t_i)\underset{\sim}{w}_d^T(t_j)] = Q_d(t_i)\delta_{ij} \tag{4}$$

$$E[\underset{\sim}{v}(t_i)] = \underline{0} \tag{5}$$

$$E[\underset{\sim}{v}(t_i)\underset{\sim}{v}^T(t_j)] = R(t_i)\delta_{ij} \tag{6}$$

where δ_{ij} is the Kronecker delta function. The assumption is made that $\underset{\sim}{x}(t_0)$, $\underset{\sim}{w}_d(t_i)$, and $\underset{\sim}{v}(t_i)$ are independent for all times t_i. The $\underset{\sim}{x}(t_0)$ vector has a mean of $\hat{\underset{\sim}{x}}_0$ and a covariance of $\underline{P}_0$.

2.2.1 Continuous Parameter Space

The system parameters define the values of the matrices, Φ, B_d, G_d, H, Q_d, and R, found in Equations (1), (2), (4), and (6). Considering the case where these system parameters are uncertain parameters (unknown constants, slowly varying, or abruptly changing), the Bayesian estimator can be used to compute the following conditional density function [14:129-133]:

$$f_{\underset{\sim}{x}(t_i),\underset{\sim}{a}|\underset{\sim}{Z}(t_i)}(\underline{x},\underline{a}|\underline{Z}_i) = f_{\underset{\sim}{x}(t_i)|\underset{\sim}{a},\underset{\sim}{Z}(t_i)}(\underline{x}|\underline{a},\underline{Z}_i) * f_{\underset{\sim}{a}|\underset{\sim}{Z}(t_i)}(\underline{a}|\underline{Z}_i) \tag{7}$$

where:

$\underset{\sim}{Z}(t_i)$ is the vector of measurements from t_0 to t_i,

$$\underset{\sim}{Z}(t_i) = [\underline{z}^T(t_i), \underline{z}^T(t_{i-1}), \ldots, \underline{z}^T(t_0)]^T \tag{8}$$

and $\underline{a}$ is the uncertain p-dimensional parameter vector which is an element of A. A is a subset of R^p, where R stands for the set of real numbers and R^p is real Euclidean p-dimensional space. Assuming that $\underline{a}$ can take on any value over a continuous range, then further evaluation of the second term on the right hand side of Equation (7) leads to [14:129; 8:20-21]:

$$f_{\underset{\sim}{a}|\underset{\sim}{Z}(t_i)}(\underline{a}|\underline{Z}_i) = f_{\underset{\sim}{a}|\underline{z}(t_i),\underset{\sim}{Z}(t_{i-1})}(\underline{a}|\underline{z}_i,\underline{Z}_{i-1}) \tag{9}$$

$$= \frac{f_{\underline{a},\underline{z}(t_i)|\underset{\sim}{Z}(t_{i-1})}(\underline{a},\underline{z}_i|\underline{Z}_{i-1})}{f_{\underline{z}(t_i)|\underset{\sim}{Z}(t_{i-1})}(\underline{z}_i|\underline{Z}_{i-1})} \tag{10}$$

$$= \frac{f_{\underline{z}(t_i)|\underline{a},\underset{\sim}{Z}(t_{i-1})}(\underline{z}_i|\underline{a},\underline{Z}_{i-1})\, f_{\underline{a}|\underline{z}(t_{i-1})}(\underline{a}|\underline{Z}_{i-1})}{\int_A f_{\underline{z}(t_i)|\underline{a},\underset{\sim}{Z}(t_{i-1})}(\underline{z}_i|\underline{a},\underline{Z}_{i-1})\, f_{\underline{a}|\underline{z}(t_{i-1})}(\underline{a}|\underline{Z}_{i-1})\, d\underline{a}} \tag{11}$$

Equation (11) could conceptually be solved recursively. Using the fact that $f_{\underline{z}(t_i)|\underline{a},\underline{z}(t_{i-1})}(\underline{z}_i|\underline{a},\underline{Z}_{i-1})$ is Gaussian with a mean of $H(t_i)\hat{\underline{x}}(t_i^-)$ and a covariance of $[H(t_i)P(t_i^-)H^T(t_i)+R(t_i)]$, where $\hat{\underline{x}}(t_i^-)$ is the conditional mean and $P(t_i^-)$ is the conditional covariance of $\underline{x}(t_i)$ just prior to the measurement taken at t_i, and starting from an a priori density function of $f_{\underline{a}}(\underline{a})$, Equation (11) can be iterated to produce $f_{\underline{a}|\underline{z}(t_i)}(\underline{a}|\underline{Z}_i)$. Furthermore, a

state estimate can be produced as the conditional mean [14:129, 8:21-22]:

$$E[\underset{\sim}{x}(t_i)|\underset{\sim}{Z}(t_i) = \underset{\sim}{Z}_i] = \int_{-\infty}^{\infty} \underset{\sim}{x}\, f_{\underset{\sim}{x}(t_i)|\underset{\sim}{Z}(t_i)}(\underset{\sim}{x}|\underset{\sim}{Z}_i)\underset{\sim}{dx} \tag{12}$$

$$= \int_{-\infty}^{\infty} \underset{\sim}{x}\, [\int_{A} f_{\underset{\sim}{x}(t_i),\underset{\sim}{a}|\underset{\sim}{Z}(t_i)}(\underset{\sim}{x},\underset{\sim}{a}|\underset{\sim}{Z}_i)\underset{\sim}{da}]\underset{\sim}{dx} \tag{13}$$

$$= \int_{-\infty}^{\infty} \underset{\sim}{x}\, [\int_{A} f_{\underset{\sim}{x}(t_i)|\underset{\sim}{a},\underset{\sim}{Z}(t_i)}(\underset{\sim}{x}|\underset{\sim}{a},\underset{\sim}{Z}_i) * f_{\underset{\sim}{a}|\underset{\sim}{Z}(t_i)}(\underset{\sim}{a}|\underset{\sim}{Z}_i)\underset{\sim}{da}]\underset{\sim}{dx} \tag{14}$$

$$= \int_{A} [\int_{-\infty}^{\infty} \underset{\sim}{x}\, f_{\underset{\sim}{x}(t_i)|\underset{\sim}{a},\underset{\sim}{Z}(t_i)}(\underset{\sim}{x}|\underset{\sim}{a},\underset{\sim}{Z}_i)\underset{\sim}{dx}] * f_{\underset{\sim}{a}|\underset{\sim}{Z}(t_i)}(\underset{\sim}{a}|\underset{\sim}{Z}_i)\underset{\sim}{da} \tag{15}$$

where the term in the brackets of Equation (15) is the estimate $\hat{\underset{\sim}{x}}(t_i^+)$ of $\underset{\sim}{x}(t_i)$ based on a particular value of the parameter vector. Based on an assumed realization of the parameter vector, the output a standard Kalman filter is this conditional expectation. The bank requires an infinite number of filters when $\underset{\sim}{a}$ is continuous over A. Likewise, using the continuous parameter space with the integrations of Equations (11) and (15) "...make this estimate computationally infeasible for online usage." [14:130].

2.2.2 Discretized Parameter Space

Discretization of the parameter space reduces the number of filters required to a finite value. When Equations (11) and (15)

are evaluated over a discrete parameter space, the integrals become summations. The probability that the k^{th} elemental filter (based on the assumption that $\underline{a} = \underline{a}_k$, one of the discrete parameter values) is correct, conditioned on the measurement history, is $p_k(t_i)$. In a development similar to that of Equation (11), this probability satisfies the following relationship [14:131; 8:23]:

$$p_k(t_i) = \frac{f_{\underset{\sim}{z}(t_i)|\underline{a},\underset{\sim}{z}(t_{i-1})}(\underline{z}_i \mid \underline{a}_k, \underline{Z}_{i-1}) * p_k(t_{i-1})}{\sum_{j=1}^{K} f_{\underset{\sim}{z}(t_i)|\underline{a},\underset{\sim}{z}(t_{i-1})}(\underline{z}_i \mid \underline{a}_j, \underline{Z}_{i-1}) * p_j(t_{i-1})} \tag{16}$$

$$\hat{\underline{x}}(t_i^+) = E[\underset{\sim}{x}(t_i) \mid \underset{\sim}{Z}(t_i) = \underline{Z}_i] \tag{17}$$

$$= \sum_{k=1}^{K} \hat{\underline{x}}_k(t_i^+) * p_k(t_i) \tag{18}$$

where, for Equations (17) and (18), a Kalman filter which assumes the parameter vector equals $\underline{a}_k$ produces the state estimate $\hat{\underline{x}}_k(t_i^+)$. Refer to Figure 1 for a pictorial representation of Equations (16), (17), and (18).

The k^{th} Kalman filter is propagated from measurement time t_{i-1} to measurement time t_i by [13:217]:

$$\hat{\underline{x}}_k(t_i^-) = \Phi_k(t_i, t_{i-1})\hat{\underline{x}}_k(t_{i-1}^+) + B_{dk}(t_i)\underline{u}(t_i) \tag{19}$$

$$P_k(t_i^-) = \Phi_k(t_i, t_{i-1})P_k(t_{i-1}^+)\Phi_k^T(t_i, t_{i-1}) + Q_{dk}(t_i) \tag{20}$$

and when the measurement, $\underset{\sim}{z}_i$, for the k^{th} filter becomes available at time t_i, the k^{th} Kalman filter is updated by [13:217]:

$$K_k(t_i) = P_k(t_i^-)H_k^T(t_i)[H_k(t_i)P_k(t_i^-)H_k^T(t_i) + R(t_i)]^{-1} \quad (21)$$

$$\hat{\underset{\sim}{x}}_k(t_i^+) = \hat{\underset{\sim}{x}}_k(t_i^-) + K_k(t_i)[\underset{\sim}{z}_i - H_k(t_i)\hat{\underset{\sim}{x}}_k(t_i^-)] \quad (22)$$

$$P_k(t_i^+) = P_k(t_i^-) - K_k(t_i)H_k(t_i)P_k(t_i^-) \quad (23)$$

where the initial conditions are given by [13:271]:

$$\hat{\underset{\sim}{x}}(t_0) = \hat{\underset{\sim}{x}}_0 \quad (24)$$

$$P(t_0) = P_0 \quad (25)$$

Note that the steady-state constant-gain Kalman filters are used in this thesis research because the transients are short and can be ignored.

As shown in Equation (18), each separate Kalman filter must have an associated probability weighting factor. This factor is calculated from Equation (16), using [14:132; 8:23-24]:

$$f_{\underset{\sim}{z}(t_i)|\underset{\sim}{a},\underset{\sim}{Z}(t_{i-1})}(\underset{\sim}{z}_i|\underset{\sim}{a}_k,\underset{\sim}{Z}_{i-1}) = 1/\{(2\pi)^{m/2}|A_k(t_i)|^{\frac{1}{2}}\} * \exp[-\tfrac{1}{2}\underset{\sim}{r}_k^T(t_i)A_k^{-1}(t_i)\underset{\sim}{r}_k(t_i)] \quad (26)$$

where

$$A_k(t_i) = H_k(t_i)P_k(t_i^-)H_k^T(t_i) + R_k(t_i) \quad (27)$$

$$\underset{\sim}{r}_k(t_i) = \underset{\sim}{z}_i - H_k(t_i)\hat{\underset{\sim}{x}}_k(t_i^-) \quad (28)$$

m = the number of measurements

The k^{th} elemental filter readily produces the residual itself, $\underline{r}_k(t_i)$, and the residual covariance, $A_k(t_i)$. The parameter estimate and covariance are found by [14:132-133; 8:24]:

$$\hat{\underline{\underline{a}}}(t_i) = E[\underline{\underline{a}}|\underline{\underline{Z}}(t_i) = \underline{Z}_i] = \int_{-\infty}^{\infty} \underline{a}\, f_{\underline{\underline{a}}|\underline{\underline{Z}}(t_i)}(\underline{a}|\underline{Z}_i)\underline{da} \tag{29}$$

$$= \int_{-\infty}^{\infty} \underline{a} \left[\sum_{k=1}^{K} p_k(t_i)\,\delta(\underline{a} - \underline{a}_k)\right]\underline{da} \tag{30}$$

$$= \sum_{k=1}^{K} a_k\, p_k(t_i) \tag{31}$$

and

$$E\{[\underline{\underline{a}} - \hat{\underline{\underline{a}}}(t_i)][\underline{\underline{a}} - \hat{\underline{\underline{a}}}(t_i)]^T|\underline{\underline{Z}}(t_i) = \underline{Z}_i]\}$$

$$= \sum_{k=1}^{K} [\underline{a}_k - \hat{\underline{\underline{a}}}(t_i)][\underline{a}_k - \hat{\underline{\underline{a}}}(t_i)]^T * p_k(t_i) \tag{32}$$

The state estimate has an error covariance of [14:131; 8:24-25]:

$$P(t_i^+) = E\{[\underline{\underline{x}} - \hat{\underline{\underline{x}}}(t_i^+)][\underline{\underline{x}} - \hat{\underline{\underline{x}}}(t_i^+)]^T|\underline{\underline{Z}}(t_i) = \underline{Z}_i\} \tag{33}$$

$$= \int_{-\infty}^{\infty} [\underline{x} - \hat{\underline{\underline{x}}}(t_i^+)][\underline{x} - \hat{\underline{\underline{x}}}(t_i^+)]^T f_{\underline{\underline{x}}(t_i)|\underline{\underline{Z}}(t_i)}(\underline{x}|\underline{Z}_i)\underline{dx} \tag{34}$$

$$= \sum_{k=1}^{K} p_k(t_i) \int_{-\infty}^{\infty} [\underline{x} - \hat{\underline{\underline{x}}}(t_i^+)][\underline{x} - \hat{\underline{\underline{x}}}(t_i^+)]^T * f_{\underline{\underline{x}}(t_i)|\underline{\underline{a}},\underline{\underline{Z}}(t_i)}(\underline{x}|\underline{a}_k, \underline{Z}_i)\underline{dx} \tag{35}$$

$$= \sum_{k=1}^{K} p_k(t_i)\{P_k(t_i^+) + [\hat{\underline{\underline{x}}}_k(t_i) - \hat{\underline{\underline{x}}}(t_i^+)][\hat{\underline{\underline{x}}}_k(t_i) - \hat{\underline{\underline{x}}}(t_i^+)]^T\} \tag{36}$$

where the covariance of the state estimate computed by the k^{th} elemental filter is $P_k(t_i^+)$.

2.2.3 Filter Convergence and Divergence

If the true value of the parameter is nonvarying, the Bayesian MMAE is optimal and converges [8:25]. For the Bayesian case, convergence [14:133] occurs when the probability associated with one elemental filter is one and the probability associated with the other elemental filters is zero. The parameter associated with the single elemental Kalman filter having probability one cannot be any arbitrary parameter. This parameter must be the parameter "nearest" the true parameter, or the state estimates may diverge (even while the parameter estimates have "converged" on this arbitrary, but erroneous, parameter). When MMAE convergence occurs [2:7-8]:

> The MMAE converges to the elemental filter with parameter value equal to, or most closely representing, the true parameter set.... [8:25]

No complete theoretical results exist for the varying parameter case [2:18; 1:8; 4:20; 8:25]. In fact, the estimator's ability to converge to a single elemental filter gives rise to some concern.

Because the filter is almost invariably based on an erroneous (or incomplete) model [14:23], the MMAE algorithm may cause the filter to converge to the wrong parameter point. Investigations of unknown biases in the measurement process have shown that the algorithm may converge to a parameter point that

is not even close to the true value in the parameter space [2:17; 1:8].

Adding dynamics pseudonoise to the assumed model in each elemental filter is one method of preventing state divergence [8:26; 14:25]. The difference between good and bad filter models can be masked by too much pseudonoise addition [8:26; 14:133]. The MMAE algorithm performance is:

> ...dependent upon a significant difference between the residual characteristics in the "correct" and the "mismatched model" filters. [8:133]

Equations (16) and (26) show that the weighted probability of a single elemental filter associated with the smallest $|A_k|$ will increase if the residuals are consistently of the same magnitude. However, the value of $|A_k|$ is independent of both the residuals and the "correctness" of the filter model, causing an erroneous result if this increase were to happen. [14:133]

One area of concern is a problem known as filter parameter divergence [1:9; 8:25] that can occur when true parameter value is actually slowly time varying. This problem occurs because the MMAE algorithm can assume a particular element filter is correct with probability equal to one. This filter produces estimates for the assumed true parameter value while the actual parameter value might move away from that value. When filter divergence occurs, the true parameter value can eventually drift far enough away from the filter-assumed parameter value that a significant

difference in estimated and true state values results [8:25]. Filter divergence can cause converging and locking onto the "wrong" filter [8:25]. Since the parameters can vary, the "right" filter at one time may very well be the "wrong" filter at some other time. The idea is never to allow the filter to "lock out" an elemental filter. A certain elemental filter may be "wrong" at first, but the parameters may vary, causing it to become the "right" filter.

The problem previously described is caused when the filter "locks on" to a point in the parameter space that is assumed to be "correct". When filter "lock on" occurs, a single elemental Kalman filter is given a probability of one (assumed to identify the parameters correctly) while the other elemental filters have zero probabilities. Given filter "lock on", the $p_k(t_{i-1})$ of Equation (16) equals zero for all the elemental Kalman filters assumed "incorrect". As seen by the equation, the $p_k(t_i)$ for these "incorrect" elemental filters cannot take on a value other than zero once the $p_k(t_{i-1})$ equals zero. This means that no matter how the true parameters vary, the MMAE algorithm will continue to produce estimates from that one elemental filter with the probability of one. Filter "lock on" causes the moving-bank MMAE to lose its ability to produce state estimates adaptively.

One method of avoiding the problem of filter "lock on" is to put a lower bound on the probabilities for each elemental filter.

By imposing a lower bound, the probabilities of each elemental filter cannot reach zero (thereby giving the probability of the "correct" filter a probability of one). This lower bound fixing is done by resetting the computed value of any probability to some minimum value (determined by prior performance analysis) if it should fall below a certain threshold. Once this reset occurs, all the probabilities are rescaled so the sum remains equal to one [2:7-9; 16:1875; 8:26-27].

An additional method is to check for state divergence among all the implemented elemental filters by means of residual monitoring (described in Sections 2.3.2 and 2.3.3). The state estimates generated by the filter with the least state divergence would then be used to "restart" the divergent elemental filters in the MMAE process. In this thesis, the method of giving lower probability bounds to the elemental filters is used. For this thesis research, the filter "lock on" is prevented by only using the method of lower probability bounding, but the use of both methods is preferable.

2.3 Moving-Bank Algorithm Development

As noted previously, the fixed-bank MMAE is too computationally burdensome for most practical considerations [1; 2; 4; 5; 8; 9; 16]. Maybeck and Hentz [16] showed that the full set of elemental filters within the fixed-bank MMAE could "be replaced by a subset of filters based on discrete parameter

values 'closest' to the current estimate of the parameter vector." [8:27]. This is done by replacing the fixed-bank MMAE with a moving-bank MMAE. Instead of implementing an elemental Kalman filter for each parameter in the parameter space, only a small subset of elemental Kalman filters is implemented in a moving bank. The probability of the non-implemented Kalman filters is now conceptually set to zero, and the probability weightings are distributed among the implemented filters of the moving bank.

When the parameter estimate changes, the filters "closest" to the new parameter are implemented while the filters "farthest" away are removed from implementation (see Figure 5 in Chapter 1). Maybeck and Hentz [2; 16] also changed the discretization levels of the moving-bank filter model. When in acquisition, coarse discretization was used for the implemented filters. As the parameter estimate improved, a finer discretization was used. Figure 6 (in Chapter 1) describes this process. In Figure 6, the bank is expanding rather than contracting, but the bank can contract as well as expand. If the varying parameter were to "jump" out of a finely discretized (contracted) bank's range (as due to the truss failing due to fatigue), the bank would expand to locate the new parameter better. Once the new parameter is located, the bank contracts around that parameter to produce more accurate state estimates. Consequently, adjacent discrete parameter space points are not necessarily occupied by

implemented filters at all times, as would be the case for the fixed-bank implementation. [8:27]

2.3.1 Weighted Average

Only the subset of filters used in the moving-bank implementation are weighted and summed. This weighting and summing is done in a similar manner as in Equations (16) and (18). Equation (18) is changed by implementing only the current J filters [8:28]:

$$\hat{\underline{x}}(t_i^+) = \sum_{j=1}^{J} \hat{\underline{x}}_j(t_i^+)\ p_j(t_i) \tag{37}$$

In the same manner, Equation (16) is changed [8:28]:

$$p_j(t_i) = \frac{f_{\underline{z}(t_i)|\underline{a},\underline{Z}(t_{i-1})}(\underline{z}_i|\underline{a}_j,\underline{Z}_{i-1})\ p_j(t_{i-1})}{\sum_{k=1}^{J} f_{\underline{z}(t_i)|\underline{a},\underline{Z}(t_{i-1})}(\underline{z}_i|\underline{a}_k,\underline{Z}_{i-1})\ p_k(t_{i-1})} \tag{38}$$

2.3.2 Moving the Bank

The decision logic for bank motion of the moving-bank MMAE estimator is extremely important. This significance is due to the facts that the moving-bank MMAE is typically not initially centered on the true parameter point and the parameter point may change. Past research has investigated several bank motion decision logics for a moving-bank MMAE estimation algorithm. These algorithms include Residual Monitoring, Parameter Position

Estimate Monitoring, Parameter Position and Velocity Estimate Monitoring, and Probability Monitoring. [2:22-26; 8:29]

Residual Monitoring

This section defines residual monitoring. First, for each elemental filter, define a likelihood quotient, $L_j(t_i)$, as the quadratic found in the exponential of Equation (26):

$$L_j(t_i) = \underline{r}_j^T(t_i) A_j^{-1}(t_i) \underline{r}_j(t_i) \tag{39}$$

If at time t_i, $\min[L_1(t_i), L_2(t_i), \ldots, L_j(t_i)] \geq T$ (where T is some threshold level having a numerical value determined during some prior performance evaluation), then the decision is made to move the bank. All the likelihood quotients are expected to exceed the threshold value (T) if the true parameter vector value lies outside the movable bank of elemental filters. The elemental filter with the smallest $L_j(t_i)$ is expected to be closest to the true parameter, and the bank is moved in the direction of the parameter space corresponding to this filter. Even though this method gives a quick response to a real need to move the bank, it also provides erroneous results for large residuals in a single instance, due possibly to a large realization of corruptive noise. [2:22-23; 8:30]

Probability Monitoring

Except that this method monitors the conditional hypothesis probabilities generated by Equation (16) rather than the residuals, this method is essentially the same as residual monitoring. Equation (16) also embodies information about the recent history of the residuals, not just the single current set of residuals, and is therefore less prone to errors due to individual badly noise-corrupted measurements. The movable bank is centered on the elemental filter which has a conditional hypothesis probability larger than a previously determined threshold. Maybeck and Hentz found that probability monitoring and parameter position estimate monitoring provided the best performance [16:1879]. Fewer computations were required, however, by probability monitoring. [2:24; 8:30].

Parameter Position Estimate Monitoring

In this method, the bank is centered around the most recent estimate of the true parameter set. This estimate is given by:

$$\hat{\underline{a}}(t_i) = \sum_{j=1}^{J} a_j \, p_j(t_i) \tag{40}$$

where J is the number of implemented filters in the moving-bank MMAE. When the bank is not centered on the point closest to the most recent true parameter set estimate, the bank is moved. [2:23-24; 8:31]

Parameter Position and Velocity Estimate Monitoring

Using a history of parameter position estimates, the "velocity" of the parameter position is estimated by this method. Using this estimated velocity, the parameter position at the next sample time is estimated. Therefore, some predictive "lead" is incorporated into the parameter position value. Maybeck and Hentz found that parameter position and velocity estimate monitoring provided acquisition times that were no faster than probability monitoring or parameter position estimate monitoring, and it tended to "lose lock" once the parameters had been correctly identified [16:1880]. [2:24; 8:31]

2.3.3 Bank Contraction and Expansion

The elemental filters of the moving-bank MMAE do not have to be adjacent to each other. The filters can be coarsely or finely discretized (see Figure 6). Even though the accuracy of the initial estimate is decreased, the probability that the true parameter set lies within the bank is increased. This enhances initial convergence. Hentz and Maybeck [2:26; 16:1877-1879] showed that parameter acquisition performance is improved when the bank starts with a coarse discretization (encompassing the entire parameter space) and then contracts to a finer discretization when the parameter covariance, Equation (32), drops below some previously determined threshold. [2:26-27; 8:33]

Another reason for expanding the bank would be to encompass a true parameter value which undertook a jump change to a value outside the present bank range. Residual or probability monitoring can detect this jump change. For residual monitoring, the likelihood ratios for all the implemented filters would exceed some previously determined threshold. For probability monitoring, the conditional hypothesis probabilities would be close in magnitude. (To confirm this, a performance analysis must be accomplished for a given application.) After this initial expansion, the bank would contract in the same manner as previously described. [2:26-27; 8:35]

2.3.4 Initialization of New Elemental Filters

New elemental filters must be implemented and old filters discarded whenever the movable bank is moved, expanded, or contracted. The values of Φ_j, B_{dj}, H_j, and K_j (steady state Kalman filter gain matrix) for the j^{th} elemental filter can be predetermined because the system model is assumed linear and time invariant for this research. However, $\hat{\underline{x}}_j(t_i^+)$ and $p_j(t_i)$ must be calculated online for each elemental filter.

The problem of finding a value for the new elemental filter's $\hat{\underline{x}}_j(t_i^+)$ is easily solved by setting it to the present moving bank estimate, $\hat{\underline{x}}(t_i^+)$. Finding $p_j(t_j)$ is a little more difficult since its value depends upon the number of new elemental filters being implemented. Either three or five new

filters are implemented if the bank should slide, as shown in Figure 4 (in Chapter 1). Redistribution of the probability weighting can be done in one of two ways: equally among the new filters or in such a way that indicates the new filter's "correctness". For either way, the sum of all the probabilities must still be unity. Hentz suggested a "correctness" method of redistributing the probabilities [2:29] which has been shown to provide no significant performance improvement over the equal probability weighting method and requires additional computations [2:104]. Therefore, for this thesis research, the probabilities will be redistributed equally among the new filters when the bank moves. [8:36]

Expanding or contracting the bank as shown in Figure 6 can result in implementing as many as nine new elemental filters. When all new elemental filters are implemented, the probability weighting is divided equally among them. [8:37]

2.4 Stochastic Controller Design

Both the moving-bank and fixed-bank MMAE algorithms can be used with several stochastic controller designs. The "assumed certainty equivalence design" methodology is used with all the controller and estimator designs of this research [15:241]. This design implements an estimator in cascade with a deterministic full-state feedback optimal controller. Independence between the estimator and full-state feedback controller design is assumed,

and the design results in the optimal stochastic controller for a linear system driven by white Gaussian noise with a quadratic cost criterion [15:17]. [8:37]

The estimator used in this thesis research is the moving-bank MMAE. All systems are assumed to be linear and time invariant with stationary noise. The elemental filter estimate is propagated by Equations (19) and (20) and updated by Equations (21) - (23) with initial conditions given by Equations (24) and (25). The association with a particular point in the parameter space $\underline{a}_k$ is given by the subscript "k". [2:19-21; 8:38]

Each controller is a linear, quadratic cost, (LQ) full-state feedback optimal deterministic controller. The cost weighting matrices in the quadratic cost are constant, and the error state space formulation is time invariant [13:29]. The cost equation to be minimized is of the form:

$$J = E\{\sum_{i=1}^{N} \tfrac{1}{2}[\underline{x}^T(t_i)X\underline{x}(t_i) + \underline{u}^T(t_i)U\underline{u}(t_i)] + \tfrac{1}{2}\underline{x}^T(t_{N+1})X_f\underline{x}(t_{N+1})\} \quad (41)$$

where:

J = cost function to be minimized

X = state weight matrix

X_f = final state weight matrix

U = control weight matrix

N = number of sample periods from t_0 to t_N

t_{N+1} = final time

t_N = last time a control is applied and held constant over the next sample period

The full development of the LQ controller is shown in Appendix A. The controllers are constant-gain in steady-state, with the gains of each controller calculated for each point in the parameter space. [8:39]

This thesis research used the multiple model adaptive controller (MMAC) approach [15:253]. For MMAC, an elemental controller is implemented for each point in the parameter space. Therefore, each of the elemental Kalman filters in the moving bank has an associated controller. Figure 2 is a pictorial representation of this algorithm. The control input is calculated by probabilistically weighting the individual control outputs as shown by:

$$\underline{u}(t_i) = \sum_{j=1}^{J} p_j(t_i)\underline{u}_j(t_i) \tag{42}$$

The individual control outputs are calculated by [8:40]:

$$\underline{u}_j(t_i) = -G_c^*[\underline{a}_j]\ \hat{\underline{x}}_j(t_i^+) \tag{43}$$

where $-G_c^*[\underline{a}_j]$ is the optimal deterministic controller gain for the assumed parameter value $\underline{a}_j$.

2.5 Summary

This chapter developed the algorithms for the moving-bank multiple model adaptive estimator and controller. The moving-bank MMAE should provide a large computational savings over the fixed-bank MMAE when both are based on the same level of parameter space discretization. The next chapter develops the large flexible space structure as a two-bay truss model and its system equations.

III. Rotating Two-Bay Truss Model

3.1 Introduction

The large flexible space structure used in this thesis is modeled as a two-bay truss, and its system equations are developed in this chapter. Rigid body rotation and bending mode dynamics are incorporated in this research by allowing the truss to rotate about a fixed point. This flexible space structure model is identical to the model used by Karnick [4:39-58] and Lashlee [8:47-69].

3.2 Second Order and State Space Model Forms

The large space structure is excited with a vibration, is actively controlled, and has n frequency modes. The following second-order differential equation describes such a system [8:47-48; 11]:

$$M\ddot{\underline{r}}(t) + C\dot{\underline{r}}(t) + K\underline{r}(t) = \underline{F}_1(\underline{u},t) + \underline{F}_2(t) \qquad (44)$$

where:

M = constant n by n mass matrix

C = constant n by n damping matrix

K = constant n by n stiffness matrix

$\underline{r}(t)$ = n dimensional vector representing the structure's physical coordinates

$\underline{F}_1(u,t)$ = the control input

$\underline{F}_2(t)$ = disturbances

The flexible space structure's control system is comprised of a set of discrete actuators with locations given in Section 3.4.4. By modelling the external disturbances as white Gaussian noise, Equation (44) becomes [8:48]:

$$M\ddot{\underset{\sim}{r}}(t) + C\dot{\underset{\sim}{r}}(t) + K\underset{\sim}{r}(t) = -b\underset{\sim}{u}(t) - g\underset{\sim}{w} \qquad (45)$$

where "$\underset{\sim}{\ }$" denotes a vector stochastic process and:

$\underset{\sim}{u}(t)$ = actuator input vector of dimension r

b = n by r matrix which identifies the position and velocity relationship between the actuators and controlled variables

$\underset{\sim}{w}$ = driving noise vector of dimension s (where s is the number of noise inputs) and is a zero mean, white Gaussian noise of strength Q

g = n by s matrix which identifies the position and the relationships between the dynamics driving noise and the controlled variables

Equation (44) can be represented in state space as [8:48]:

$$\dot{\underset{\sim}{x}} = F\underset{\sim}{x} + B\underset{\sim}{u} + G\underset{\sim}{w} \qquad (46)$$

where $\underset{\sim}{x}$ is 2n-dimensional and given by:

$$\underset{\sim}{x} = \begin{bmatrix} \underset{\sim}{r} \\ \dot{\underset{\sim}{r}} \end{bmatrix} \qquad (47)$$

and the open-loop plant matrix F, the control matrix B, and the noise matrix G are [8:49]:

$$F = \begin{bmatrix} 0 & I \\ -M^{-1}K & -M^{-1}C \end{bmatrix}_{2n \text{ by } 2n} \tag{48}$$

$$B = \begin{bmatrix} 0 \\ -M^{-1}b \end{bmatrix}_{2n \text{ by } r} \tag{49}$$

$$G = \begin{bmatrix} 0 \\ -M^{-1}g \end{bmatrix}_{2n \text{ by } s} \tag{50}$$

By letting the noise enter the system at the same location as the actuators, the b matrix equals the g matrix and r equals s. The measurements are taken by accelerometers and gyros with locations given in Section 3.4.4. Velocity measurements are found by integrating the accelerometer measurements once. Position measurements are found by integrating the accelerometer measurements twice. Because the same accelerometer measurements are used for both position and velocity, the position and velocity measurements are co-located. This would imply that the measurement noise matrix (R) is <u>not</u> diagonal, since the position and velocity "noises" would be highly correlated. However, for this thesis effort, the measurement noise matrix <u>is</u> assumed diagonal for simplicity. Once the position and velocity measurements are found, the system model receives them. Roundoff

errors caused by the A/D conversion used during the integration process are assumed small. Since these roundoff errors are caused by the measurement process, they can be accounted for by increasing the value of the measurement noise covariance matrix (R). Sensor model deficiencies or actual external measurement noise are the causes of position and velocity measurement noise. [8:50]

The measurements are modeled by [8:50]:

$$\underset{\sim}{z} = \begin{bmatrix} H & 0 \\ 0 & H' \end{bmatrix} \underset{\sim}{x} + \underset{\sim}{v} \qquad (51)$$

m by 2n

where:

m = the number of measurements

$\underset{\sim}{v}$ = m-dimensional vector of uncertain measurement disturbance modeled as zero-mean white Gaussian noise of covariance $\underline{R}$ [13:114]

H = position measurement matrix

H'= velocity measurement matrix

For this research, the two measurement matrices, H and H', are equal to each other since the position and velocity measurements are co-located.

3.3 Modal Analysis

The system is transformed from a physical coordinate system to a modal coordinate system [18:352-353]. Decoupling is insured

by setting the damping matrix equal to a linear combination of the mass and stiffness matrices [11]:

$$C = \alpha K + \beta M \tag{52}$$

The determination of values for α and β will be shown to be unnecessary later in this section.

The modal coordinates represented by $\underline{\Omega}$ are related to the physical coordinates, represented by $\underline{r}$, by the n by n modal transformation matrix T in the equation:

$$\underline{r} = T\underline{\Omega} \tag{53}$$

The matrix T is a matrix of eigenvectors and the solution to [11; 17; 19; 20]:

$$\omega^2 MT = KT \tag{54}$$

where ω is the natural or modal frequency [3:66]. By substituting Equation (53) into Equation (44), the state representation is transformed into modal coordinates [4; 5; 8:52; 9]:

$$\dot{\underline{\underline{x}}}' = F'\underline{\underline{x}}' + B'\underline{u} + G'\underline{\underline{w}} \tag{55}$$

where $\underset{\sim}{x}'$ is defined as:

$$\underset{\sim}{x}' = \begin{bmatrix} \underset{\sim}{\Omega} \\ \dot{\underset{\sim}{\Omega}} \end{bmatrix}_{2n \text{ by } 1} \tag{56}$$

and the transformation of the matrices given in Equations (48) - (50) gives the matrices F', B', and G':

$$F' = \begin{bmatrix} 0 & I \\ -T^{-1}M^{-1}KT & -T^{-1}M^{-1}CT \end{bmatrix}_{2n \text{ by } 2n} \tag{57}$$

$$B' = \begin{bmatrix} 0 \\ -T^{-1}M^{-1}b \end{bmatrix}_{2n \text{ by } r} \tag{58}$$

$$G' = \begin{bmatrix} 0 \\ -T^{-1}M^{-1}b \end{bmatrix}_{2n \text{ by } r} \tag{59}$$

It is now helpful to write the F' matrix as [19; 20]:

$$\underset{\sim}{F}' = \begin{bmatrix} 0 & I \\ [-\omega_i^2] & [-2\zeta_i\omega_i] \end{bmatrix}_{2n \text{ by } 2n} \tag{60}$$

where ω_i is the undamped natural frequency of the i^{th} mode, and ζ_i is the damping ratio for the same mode. After the modal transformation, the measurement equation becomes [8:53]:

$$\underset{\sim}{z} = \begin{bmatrix} HT & 0 \\ 0 & HT \end{bmatrix} \underset{\sim}{x}' + \underset{\sim}{v} \tag{61}$$

Because the system formulation is done in modal coordinates, the damping throughout the structure is assumed to be uniform [11]. By selecting the damping coefficient, ζ_i, in Equation (60), the level of structural damping is determined. The calculation of the natural (or modal) frequency, ω_i, does not depend on the value of ζ_i selected. Assuming uniform damping throughout the entire structure provides better physical insight into the formulation of the problem and simplifies the approach [8:53]. For these reasons, Equation (60) is used to determine the F' matrix rather than Equation (52). In this manner, it is unnecessary to find the values of α and β as previously mentioned. Since a damping coefficient of $\zeta_i = 0.005$ is characteristic of large space structure damping [11; 8:53], it is chosen for this thesis research.

3.4 Two-Bay Truss

For this thesis research, the two-bay truss developed by Karnick [4:45-58; 5:1251] and used by Karnick and Lashlee [8:53-69; 9:9] in their thesis efforts is used. This structure is a two-bay truss which rotates about a fixed point and describes a large flexible space structure. The following sections develop the two-bay truss model, its sensors and actuators, the reduced state representation, and the parameter variations considered.

3.4.1 Background

The main goal of this truss development was to provide a structure which: "...approximated a space structure that has a hub with appendages extending from the structure" [8:54]. To reach this goal, the original two-bay truss shown in Figure 7 developed by Banda and Lynch [11] was modified by first adding non-structural masses to nodes 1 through 4 (reducing the structural frequencies to match those of a flexible space structure) and then adding rigid body motion to the truss. By making the mass of the hub large with respect to the mass of the appendage, a command can cause the hub to rotate and point the appendage in a specified direction. [8:54]

The original two-bay truss model of Figure 7 was modified to give rigid body motion by the addition of a fixed point (node 7) and allowing the rotation of the truss in the x-y plane about this fixed point as shown in Figure 8. Very large rods were used to connect the fixed point (node 7) to the two-bay truss. This connection provided a very "stiff" link between the fixed point and the truss. The purpose of this "stiff" link was to keep the lower modal frequencies of the modified two-bay truss (having the rigid body motion) close to the lower modal frequencies of the unmodified truss. The addition of this "stiff" link did, however, introduce high frequency modes into the modified truss structure. The effects of these additional high frequency modes will be investigated in this research effort. [4; 5; 8:56-57; 9]

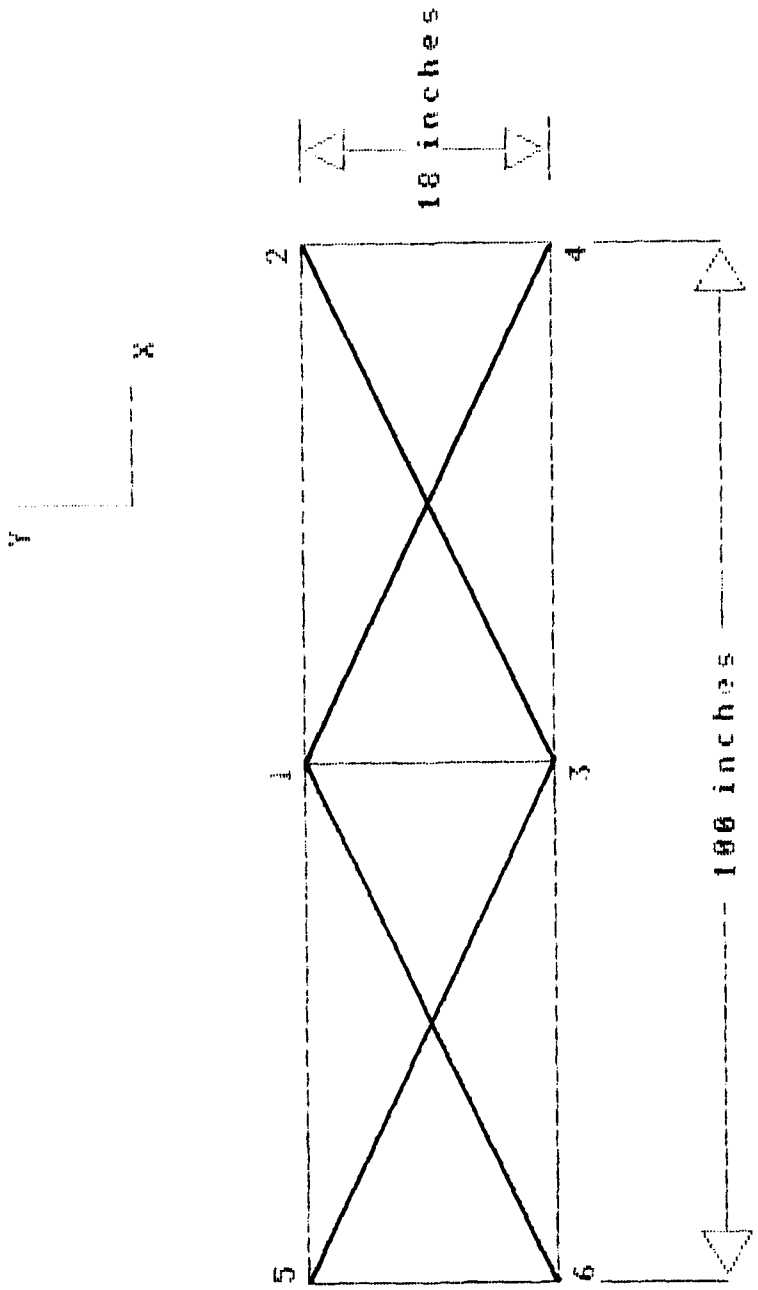

Figure 7

Original (Unmodified) Two-Bay Truss Structure

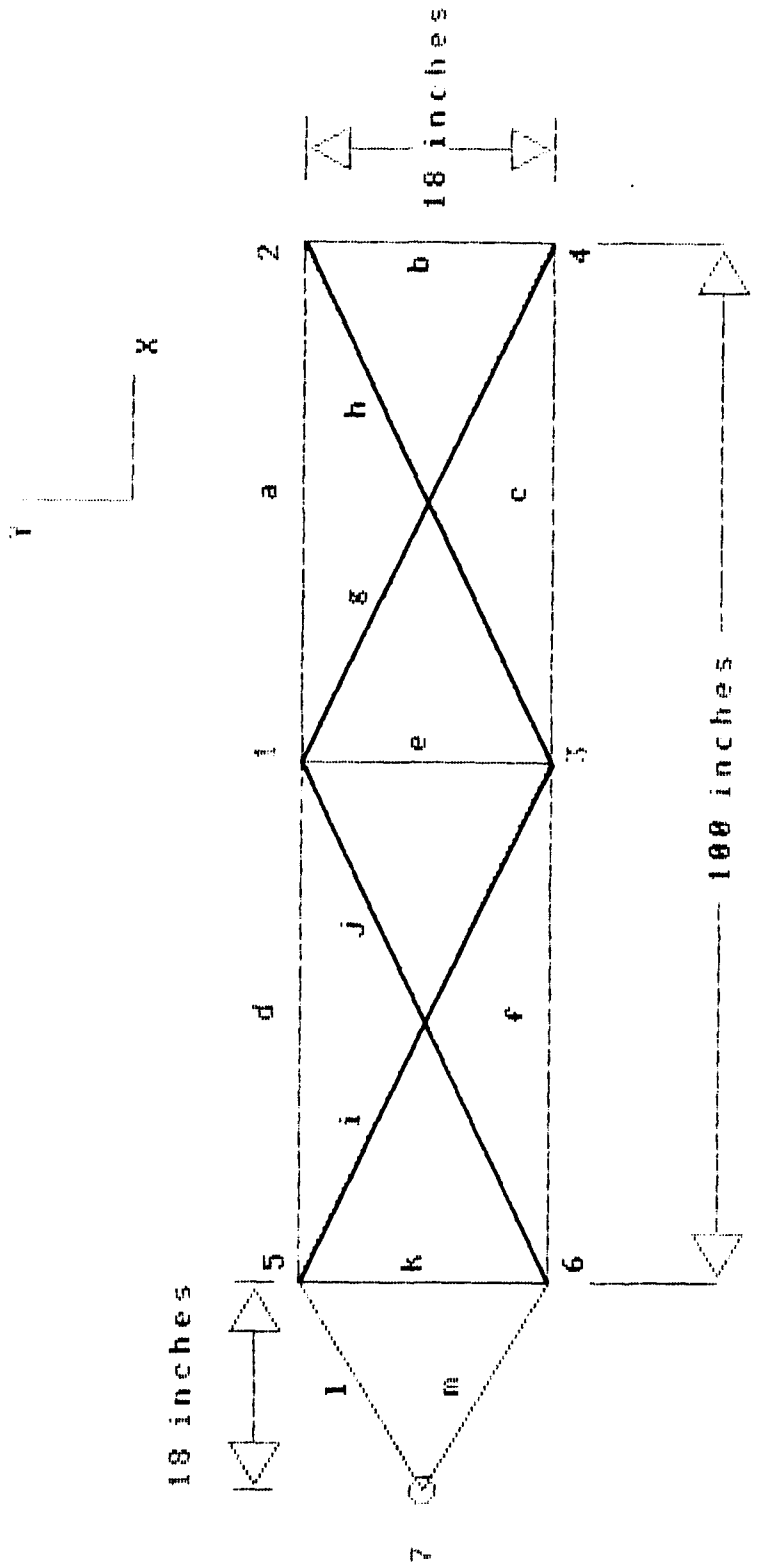

Figure 8

Modified Two-Bay Truss Structure

3.4.2 Two-Bay Truss Construction

The modified two-bay truss consists of 13 aluminum rods which each have a modulus of elasticity of 10 pounds per square inch and weight density of 0.1 pounds per inch [4:48; 8:57; 19]. Table 1 gives the cross-sectional area of each truss rod [4:48; 8:57]:

Table 1

Cross-Sectional Area of Each Structural Member

Member	Area (in^2)	Member	Area(in^2)
a	0.00321	h	0.00328
b	0.00100	i	0.00439
c	0.00321	j	0.00439
d	0.01049	k	0.20000
e	0.00100	l	0.20000
f	0.01049	m	0.20000
g	0.00328		

These member cross-sectional areas were found from a weight optimization of the unmodified truss structure (Figure 7). The goal of the design was to minimize the weight without changing the fundamental frequency so that the system's natural frequency response remained constant. Rods k through m were used as the "stiff" link between the truss and node 7. This "stiff" link was achieved by arbitrarily selecting the areas of these rods (k through m) larger than the areas of the other rods. [4:49; 8:57-58]

To achieve the low frequencies associated with large space structures, non-structural masses were located at nodes 1, 2, 3,

and 4 of the modified (Figure 8) two-bay truss [11]. An optimizing technique [20] was used to select the value of these masses at 1.294 pound-second^2 per inch. These masses produced a lowest mode frequency in the fixed two-bay truss of 0.5 Hertz [11].

Finite element analysis was used to find the mass and stiffness matrices for the system model. In finite element analysis, the structure is modeled as: "...a finite number of rods connected by elements" [8:58; 20]. A software program written by Venkayya and Tischler was used to produce these matrices [20]. A number of different elements can be used by this program. For this thesis effort, the modulus of elasticity, weight density, and cross-sectional area were used to describe the truss rod members. The mass and stiffness matrices produced by this computer program each were of a dimension equal to the truss model's number of degrees of freedom (DOF). A specific node and DOF were associated with each row of the mass and stiffness matrices. The desired result was to have the mass and stiffness matrices reduced dimensionally to 12-by-12, giving a 24-state model. This was accomplished by considering only planar motion in the x-y plane, fixing node 7 (which eliminated all three DOF for this node), and associating rows of the mass and stiffness matrices with the x-axis DOF of each node. For the specified modified two-bay truss structure, the mass and stiffness matrices are given in Appendix B. These matrices were

the nominal from which variations in the parameter space were considered. [4:49-50; 8:59]

3.4.3 Sensors and Actuators

Accelerometers and actuators were placed at nodes 1 and 2 of the modified two-bay truss structure (Figure 8). The co-location of the accelerometers and actuators was not necessary but done to simplify the system model. Position and velocity measurements are produced by integrating the accelerometer outputs. The hub (node 7), at which point was located an inertial wheel as an actuator, contained an additional two co-located gyro sensors as well as that actuator. [4:50-51; 8:60]

Equations (46) through (50) provide direct physical variable formulation of the position and velocity states. Equations (55) through (59) provide direct modal formulation of the angular displacement and velocity. The H and b matrices were calculated in physical variable formulations, transformed to modal coordinates, and augmented with the appropriate zero matrices to give the proper dimensional and state results. [4:51; 8:60]

3.4.4 Physical System Parameter Uncertainty

For adaptive estimation and control to be needed, rather than nonadaptive, the system model must have some type of parameter uncertainty. By considering two physically motivated parameter variations, a 10-by-10 two dimensional discrete

parameter space was made. The mass matrix entries due to the four non-structural masses and the stiffness matrix entries are the variable parameters. Mass variations could be due to fuel tank depletion or addition, or fuel being shifted. Structural fatigue or rod failure could be the cause of the stiffness variations. The parameter discretization for this thesis is the discretization level found by Lashlee [8:116]. [8:60-61]

3.5 State Reduction

As shown previously, the mass and stiffness matrices are of dimension 12, which gives a 24-state system model. For previous thesis efforts [1; 2; 4; 5; 8; 9; 16], this 24-state system model was too large. This thesis effort will determine the effect of a 24-state truth model used in conjunction with a reduced order (6-state) filter or stochastic controller model. Because the reduced order model is used for the filter model, it is required to generate such a reduced order model. The method used for order reduction, singular perturbations [6; 7; 11; 15:219], is described in this section.

3.5.1 Development

The first step is to reformulate the deterministic system as follows [4:52-58; 8:62]:

$$\begin{bmatrix} \dot{\underline{x}}_1 \\ \dot{\underline{x}}_2 \end{bmatrix} = \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix} \begin{bmatrix} \underline{x}_1 \\ \underline{x}_2 \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} \underline{u} \tag{62}$$

$$\underline{z} = \begin{bmatrix} H_1 & H_2 \end{bmatrix} \underline{x} \tag{63}$$

The goal of the reduction scheme is to retain the low frequency states (the $\underline{x}_1$ states) and eliminate the high frequency modes (the $\underline{x}_2$ states) by assuming steady state is instantaneously reached for the high frequency modes ($\dot{\underline{x}}_2 = \underline{0}$). F_{11} and F_{12} are square matrices, and assuming F_{22}^{-1} exists, $\underline{x}_2$ can be expressed in terms of $\underline{x}_1$ [4:53; 8:62-63]:

$$\dot{\underline{x}}_2 = \underline{0} = F_{21}\underline{x}_1 + F_{22}\underline{x}_2 + B_2\underline{u} \tag{64}$$

$$\underline{x}_2 = -F_{22}^{-1}(F_{21}\underline{x}_1 + B_2\underline{u}) \tag{65}$$

and substituting this into Equations (62) and (63) yields:

$$\dot{\underline{x}}_1 = F_r\underline{x}_1 + B_r\underline{u} \tag{66}$$

$$\underline{z} = H_r\underline{x}_1 + D_r\underline{u} \tag{67}$$

where:

$$F_r = (F_{11} - F_{12}F_{22}^{-1}F_{21}) \tag{68}$$

$$B_r = (B_1 - F_{12}F_{22}^{-1}B_2) \tag{69}$$

$$H_r = (H_1 - H_2F_{22}^{-1}F_{21}) \tag{70}$$

$$D_r = (-H_2F_{22}^{-1}B_2) \tag{71}$$

The D_r matrix is a direct feed-forward term which did not exist before order reduction [8:63].

Applying this order reduction technique to the modal coordinate system, Equation (55), and reordering Equation (57) into the reduced order form, gives Equation (72) [4:54; 8:63-64]:

$$F = \left[\begin{array}{cc|cc} 0 & I & & \\ [-\omega_1^2] & [-2\zeta_1\omega_1] & & 0 \\ & 0 & & \\ \hline & & 0 & I \\ & & [-\omega_2^2] & [-2\zeta_2\omega_2] \end{array}\right]_{2n \text{ by } 2n} \tag{72}$$

where the upper partition contains the modes to be retained and the lower partition contains the modes assumed to reach steady state instantaneously. The dimension of F_{11} in Equation (62) is therefore twice the number of slow system model modes, and the dimension of F_{22} is twice the number of modes assumed to reach steady state instantaneously. Partitions F_{12} and F_{21} are shown to be zero by comparing Equation (62) with Equation (72). By examining Equations (66) and (67), the following results [4:54; 8:64]:

$$F_r = F_{11} \tag{73}$$

$$B_r = B_1 \tag{74}$$

$$H_r = H_1 \tag{75}$$

$$D_r = (-H_2 F_{22}^{-1} B_2) \tag{76}$$

All the reduced order matrices, with the exception of D_r, are found simply by truncating those states associated with $\underline{x}_2$ from the original truth model equations. D_r is dependent upon the state terms which are assumed to reach steady state instantaneously. [4:55; 8:64]

By examining the form of Equation (76), the determination of D_r is greatly simplified. First, notice how H_2 is similar in form to Equation (51) [4:55; 8:65]:

$$H_2 = \begin{bmatrix} H_c & 0 \\ 0 & H_c' \end{bmatrix} \tag{77}$$

The unmodeled position and velocity states are represented by H_c and H_c' respectively. The subscript c denotes the co-located measurements. Because the position and velocity measurements are found from the co-located accelerometers, the position and velocity measurement matrices are identical. However, the two matrices must remain distinct from each other for the reduced order matrix development as shown later by Equation (81). The matrix F_{22} is square and of the form [4:55; 8:65]:

$$F_{22} = \begin{bmatrix} 0 & I \\ [-\omega_2^2] & [-2\ \zeta_2\ \omega_2] \end{bmatrix} \tag{78}$$

Each of the four partitions is a matrix that is square and diagonal. The dimension of each partition depends on the number of retained states. F_{22} has an inverse given by [4:56; 8:66]:

$$F_{22}^{-1} = \begin{bmatrix} [-\omega_2^2]^{-1}[2\ \zeta_2\ \omega_2] & [-\omega_2^2]^{-1} \\ I & 0 \end{bmatrix} \tag{79}$$

The form of matrix B_2 is similar to that of the matrix B in Equation (49) [4:56; 8:66]:

$$B_2 = \begin{bmatrix} 0 \\ b' \end{bmatrix} \tag{80}$$

where b' corresponds to the unmodeled states associated with the matrix product $-M^{-1}b$. Evaluating Equation (76) leads to [4:56; 8:66]:

$$D_r = \begin{bmatrix} H_c[-\omega_2^2]^{-1}b' \\ 0 \end{bmatrix}_{\text{m by r}} \tag{81}$$

where m is the number of measurements and r is the number of controls. Since the lower partition of D_r is zero, only the position measurements are affected. Only the position portion and not the velocity portion of the measurement matrix affects the D_r matrix. Since $[-\omega_2^2]$ is diagonal, its inverse is easily found. Appendix B has an example of detailed system matrix development and order reduction. [4:56; 8:66]

3.5.2 Order Reduction Selection

By examining the eigenvalues and modal frequencies of the unreduced system as displayed in Table 2, the number of modes to retain are selected. The eigenvalues shown in Table 2 are for the F matrix which corresponds to the (5,5) position in the parameter space. The very lightly damped ($\zeta_i = 0.005$) large space structure system is used to select the eigenvalues. Table 2 shows that the frequencies can be divided into closely spaced groups. For example, one group of closely space frequencies is

clearly seen in modes 4, 5, and 6. The goal is to provide adequate estimation and control, and a sufficient number of frequencies must be retained to achieve this goal. In this thesis effort, the reduced order filter model has six states, composed of the first three modes in Table 2 (rigid body mode and two lowest frequency modes). The truth model has 24 states. The effect of the order mismatch between the filter model and truth model is the crux of this research. [4:57; 8:67]

Table 2
Eigenvalues and Frequencies

Mode Number	Eigenvalues			Frequencies (Hz)
1	0.00	±	j0.00	0.00
2	-0.05	±	j9.18	9.18
3	-0.12	±	j23.29	23.29
4	-0.15	±	j30.51	30.51
5	-0.16	±	j32.18	32.18
6	-0.17	±	j33.87	33.87
7	-0.28	±	j56.17	56.17
8	-0.30	±	j60.07	60.07
9	-4.83	±	j965.99	966.01
10	-44.18	±	j8836.57	8836.68
11	-56.42	±	j11283.18	11283.32
12	-97.77	±	j19553.06	19553.30

The eigenvalues and frequencies given in Table 2 differ from those presented in previous theses [4:57; 8:68]. Table 2 was generated for this thesis effort because the information contained in the previous theses about the eigenvalues and frequencies is incomplete.

3.6 Summary

This chapter developed the truth model (24-states) and the reduced order (6-state) system model for use in the moving-bank MMAE and MMAC. The system model is a two-bay truss with rigid body motion. The reduced order filter model retains the rigid body mode and the first two bending modes.

IV. Simulation Plan

4.1 Introduction

The actual space structure movement and estimation/control must be simulated to evaluate the performance of the moving-bank multiple model adaptive estimation and control. A Monte Carlo computer analysis is the method by which this simulation is performed. This chapter covers three areas: the Monte Carlo simulation background, the computer software outline, and the simulation plan.

4.2 Monte Carlo Analysis

A Monte Carlo analysis is used to obtain statistical information on the estimation and control performance of the moving-bank multiple model adaptive estimator and controller. A covariance analysis is not used because adaptive estimation and control is nonlinear and requires a full Monte Carlo analysis. A Monte Carlo analysis is defined:

> Essentially, many samples of the error process are generated by simulation and then the sample statistics computed directly. If enough samples are generated, these should approximate the process statistics very well. [13:329]

For this research, the Monte Carlo simulation involves two system models. One is a 24-state truth model and the other is a 6-state filter model. The 6-state filter model is the same as the one used by Karnick and Lashlee in their theses [4:52-58;

8:47-69]. The 24-state truth model was developed for this research. A full development of the truth model is shown in Appendix C.

As stated previously [13:329], "enough" samples must be generated during a Monte Carlo analysis to "approximate the process statistics". For this research, 10 Monte Carlo simulations are used because past research [1:52] has shown that running the Monte Carlo simulation 10 or more times is sufficient. This sufficiency was demonstrated by:

> ...observing how the computed sample statistics change as the number of runs is increased; after a sufficient number of runs, the sample statistics will converge to a constant... [1:52].

The block diagram of Figure 9 illustrates the Monte Carlo simulation [2:47-48; 1:53; 4:62; 8:73]. The undefined variables are:

$\underset{\sim}{x}_t(t_i)$ = "truth model" states

$\hat{\underset{\sim}{x}}(t_i)$ = estimates of the system states

$\hat{\underset{\sim}{a}}(t_i)$ = estimates of the uncertain parameter vector

$\underset{\sim}{e}_a(t_i)$ = error in the parameter estimate

$$\underset{\sim}{e}_a(t_i) = \underset{\sim}{a}_t(t_i) - \hat{\underset{\sim}{a}}(t_i)$$

$\underset{\sim}{e}_x(t_i)$ = error in the system state estimate

$$\underset{\sim}{e}_x(t_i) = T_x\underset{\sim}{x}_t(t_i) - \hat{\underset{\sim}{x}}(t_i)$$

T_x = n-by-n_t matrix which makes the dimensions of the truth and filter model matrices compatible. T_x is a 6-by-6 identity matrix with a 6-by-18 zero matrix augmented to the right for this investigation.

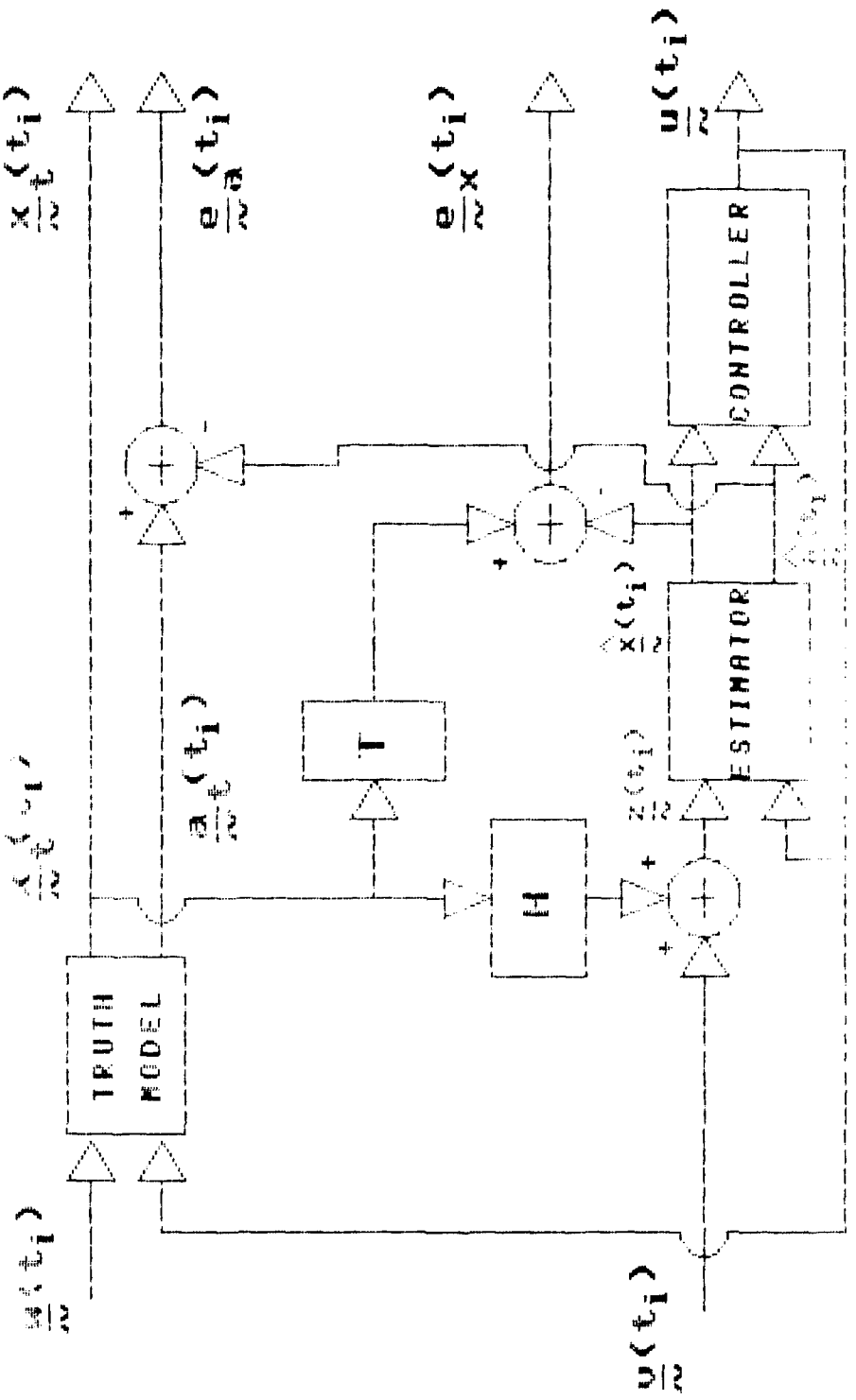

Figure 9

System Estimation and Control Simulation

Computing the sample mean and covariance of the variables of interest involves [13:129; 8:74]:

$$E[\underset{\sim}{e}_x(t_i)] \approx \underline{M}_{ex}(t_i) = (1/L)\sum_{k=1}^{L} \underline{e}_{xk}(t_i) \tag{82}$$

$$E\{[\underset{\sim}{e}_x(t_i) - \underline{M}_{ex}(t_i)][\underset{\sim}{e}_x(t_i) - \underline{M}_{ex}(t_i)]^T\} = P_{ex}(t_i) \approx$$

$$[1/(L-1)]\sum_{k=1}^{L} \underline{e}_{xk}(t_i)\underline{e}_{xk}^T(t_i) - [L/(L-1)]\underline{M}_{ex}(t_i)\underline{M}_{ex}^T(t_i) \tag{83}$$

where:

L = the total number of Monte Carlo simulation runs

$\underline{e}_{xk}(t_i)$ = the value of $\underset{\sim}{e}_x(t_i)$ during the k^{th} simulation run

Equations of the form of (82) and (83) can also be used to generate the sample mean and covariance for $\underset{\sim}{e}_a(t_i)$.

For this thesis research, the states themselves are the modal states, and there are two ways to represent the state estimation error. The first way is to represent the state estimation error by $T\underline{x}_t(t_i) - \hat{\underline{x}}(t_i)$ where T is a 6-by-24 transformation matrix. This is the option used for this thesis research. Physically, this method represents the rigid body and first two bending mode generalized coordinate positions and velocities. The state estimation error is then the error between the filter's estimated states and the <u>real</u> first three modes' (rigid body and first two bending) positions and velocities. The second way is to represent the state estimation error as physical

positions and velocities located at physical nodes, which correspond to the sensor/actuator locations, on the two-bay truss. For this option, the state estimation errors are generated by $H_t\underline{x}_t(t_i) - \hat{\underline{x}}(t_i)$ where H_t is the 6-by-24 truth model measurement matrix. The state estimation error is now the positions and velocities due to the filter-assumed 6 states compared to those same positions and velocities due to all 24 modal states of the truth model. This option would provide better insight into the actual physical two-bay truss movement and should be used in follow-on thesis work. Likewise, for future research, an evaluation of the relative magnitudes of modal coordinates included in the filter versus the magnitudes of the modal coordinates omitted from that reduced order model filter would be instructive.

The controller block of Figure 9 is replaced by a dither signal when estimation is being evaluated, to enhance identifiability above that of the uncontrolled case. Determining the appropriate amplitude and frequency of the dither signal is done by trial and error. In this thesis, the dither signal has an amplitude of 100 and a frequency of 30 Hertz. [8:74]

The evaluation of the performance of the MMAE is done by looking at the state and parameter estimates. The MMAE can be compared to other types of estimation schemes by examining the error in the state estimates. The value of $\underline{e}_x(t_i)$ in Figure 9 is

of most importance for the estimation evaluation; the value of $\underset{\sim}{e}_a(t_i)$ is of secondary importance. The values of $\underset{\sim}{x}_t(t_i)$ and $\underset{\sim}{u}(t_i)$ are of primary interest for control evaluation. The main goal of the MMAE estimator is to estimate (and control) the states rather than the parameters. Examining the parameter estimation errors can indicate the accuracy of state estimate. [8:75]

The examination of the statistics (shown in Figure 9) of the true state values provides a means of examining the estimator/ controller combination. The control objective for this research is to "point" the truss structure in a desired direction and quell out any resulting oscillations. Unrealistic control inputs can be checked by examining the control input magnitudes. This research does not limit these magnitudes, but a control input magnitude of 100 pounds is given as reasonable [8:75].

4.3 Software Description

The MMAE/MMAC simulation is performed using three computer programs written in FORTRAN on a machine using the UNIX operating system. These programs are a result of a continual development and modification process started by Hentz [2; 16] and continued by Filios [1], Karnick [4; 5], and Lashlee [8; 9]. The programs were modified in two major areas for this thesis research. First, the programs were moved off a slower CDC CYBER using the NOS operating system to a faster and more versatile UNIX

operating system on an Elxsi 6400 superminicomputer. Second, the programs were modified to produce and propagate a 24-state truth model (the previous version was limited to a 6-state truth model). The first program, known as the preprocessor [8:76], produces the required elemental Kalman filter components used in the Monte Carlo simulation. The second program, known as the primary processor, simulates the MMAE/MMAC and performs the actual Monte Carlo simulation. The third program, known as the postprocessor, calculates and plots the mean plus or minus the standard deviation (σ).

4.3.1 The Preprocessor

The preprocessor produces all the matrices (Φ, B_d, G_d, H, and D_r), the Kalman filter gains, and the LQ controller gains for each point in the parameter space. The state and control weighting matrices (X and U), the measurement noise covariance matrix (R), the dynamic driving noise matrix (Q), and the time increment (dt) are all variables entered into the preprocessor through an external input file. Likewise, the mass and stiffness matrices and the parameter space discretization level are entered into the preprocessor through the same input file. Both the 24-state truth model and 6-state filter model can be determined, once the mass and stiffness matrices' parameters are identified. [8:77]

4.3.2 Primary Processor

The actual Monte Carlo and moving-bank simulation is performed by the primary processor. The simulation performed by the primary processor involves propagating the true (24-state) system and the currently implemented elemental 6-state Kalman filters in the moving bank forward in time for each sample period. The elemental filters are updated after a noise corrupted measurement of the true system is made. Once the update is performed, the program calculates the adaptive state estimate and/or the adaptive control, and decides whether the moving bank of filters should move, expand, contract, or remain. Values of interest are written to data files once the sample period is complete.

The primary processor relies on three input files. The first input file is the 6-state filter model system generated by the preprocessor. The second input file is the 24-state truth model system generated by the preprocessor. The third input file tells the primary processor the number of Monte Carlo runs to perform, the true parameter values, the magnitude of the dither signal, the time increment (dt), information that specifies which move/contract/expand algorithms to use, initial probability weightings, and initial filter state values. The primary processor generates output data files which contain the true and estimated state values, a history of the parameter values assumed

by the center elemental Kalman filter (of the moving bank), and information necessary for the postprocessor.

4.3.3 Postprocessor

Information written to data files from the primary processor is used by the postprocessor to generate the sample mean and standard deviations of the state estimate error ($\underline{x}_t(t_i) - \hat{\underline{x}}(t_i)$) from t_0 to t_{N+1} (where N is the number of sample periods). Once this data is calculated, the postprocessor produces plots of the mean plus or minus the standard deviation (σ) of the state estimate error. A postprocessor run is required each time the primary processor is run using new input data. This information is plotted for both the state estimation and controller evaluation. A better representation of the controller performance would be plots of the sample statistics of the states themselves (demonstrating regulation of the states to zero); however, this information was not extracted for plotting purposes.

4.4 Simulation Plan

The simulation plan involves four studies. The first study is to modify Lashlee's software for the 24-state truth model generation and propagation, and to produce results similar to those found by Lashlee. The second study is a noise level determination. The third study is a robustness analysis, and the fourth study is a controller analysis. The parameter (7,6) is

used for truth model during the simulation. This parameter is used because the state and control weighting matrices were tuned for this point by Lashlee [8:143]. Ideally, each G_c^* should be tuned to the assumed parameter $\underline{a}_k$. The parameter space discretization level used throughout this thesis is the discretization level found by Lashlee in his thesis [8:115-117]. Even though it is not an explicit part of the simulation plan, moving the simulation software from the CDC Cyber to the Elxsi was a time-intensive part of this thesis research.

4.4.1 Duplication of Past Work

Because the software was moved from the CDC Cyber to the Elxsi, it was important to make sure the programs worked correctly by duplicating Lashlee's results for the open-loop estimation. The object of this study is two fold: first to ensure the original unmodified software works correctly on the Elxsi, and second to ensure the software that has been modified for the 24-state truth model works correctly. To generate stable parameter and state estimation results, Lashlee required stochastic adaptive control as described by Figure 10. For this control, the control outputs are not probabilistically weighted as in Equation (42), and the parameter estimates are used to give the optimal control, $G_c^*[\hat{\underline{a}}(t_i)]$. This control is not the same as MMAC.

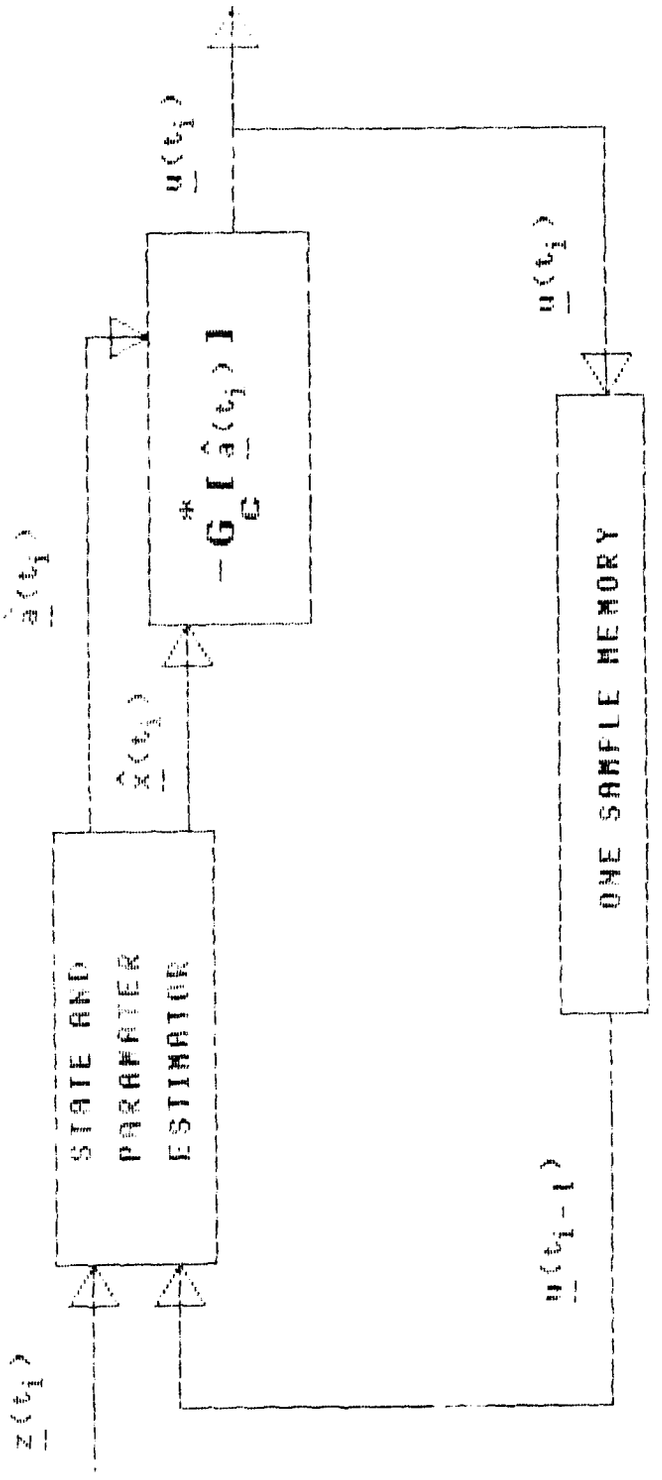

Figure 10

Stochastic Adaptive Control

The programs used are the original (unmodified) software for a 6-state filter and 6-state truth model and the (modified) software for a 24-state truth model and 6-state filter model. All software is run on the Elxsi. To duplicate Lashlee's results, the 18 unmodeled states are set to zero in the modified (24-state truth model) version of the primary processor, and the Monte Carlo simulation is run. Both the unmodified and modified versions of the software (preprocessor, primary processor, and post processor) are run. Running the unmodified software demonstrates to some degree that the program does in fact work on the Elxsi, and running the modified software demonstrates to some degree that the modifications are correct. The post processor generates the plots of the outputs. The results of this duplication effort are given in Section 5.2.1.

4.4.2 Noise Level Determination

As shown by Lashlee and Maybeck, "The values of the measurement noise covariance matrix (R) play an extremely important role in the performance of the moving-bank algorithm" [9:29]. For this reason, a noise level determination study is performed during the simulation process. The purpose of the noise level determination is to find the values of the measurement noise covariance matrix (R) and dynamics noise strength matrix (Q) which provide an MMAE that produces accurate state estimates. For this study, as well as all the following state estimation studies, no type of control is used to maintain

stability (as opposed to Lashlee's studies as previously described in Section 4.4.1). Therefore, true open-loop estimation is evaluated in these studies.

Lashlee found (8:198) that the values of the R matrix play a very large role in the moving-bank MMAE performance. Furthermore, the major difference between the truth model and reduced order model representations is the inclusion versus deletion of higher order mode effects in the measurement relations for $\underline{z}_t = H_t\underline{x}_t + \underline{v}_t$ and $\underline{z} = H\underline{x} + \underline{v}$; this difference can be compensated for by increasing R_f. Therefore, this research is initially limited to tuning <u>only</u> R_f rather than <u>both</u> R_f and Q_f. The Q matrix is the same for both the truth and filter models and is allowed to vary in a <u>sensitivity</u> analysis. A better study method would be to allow both R_f and Q_f to vary (a true filter tuning analysis), and this is done (as described later) for the controller study.

The fact that this study does not use the control Lashlee needed may cause some concern by implying that a comparison of the two thesis efforts (Lashlee's and this effort), may not be valid. This concern is outweighed by the fact that the object of this thesis effort is to evaluate <u>open-loop</u> state estimation. Since control of the type used by Lashlee was not required for the state estimation of this thesis, the control is not used and provides "true" open-loop state estimation results. These

results are still compared to Lashlee's results, but the difference in the methods of estimation are kept in mind.

The logical place to start the noise determination is with the measurement noise covariance matrices for the truth and filter models (R_t and R_f) equal to the R matrix found by Lashlee [8:94] and the dynamics noise covariance matrix (Q) equal to the Q matrix found by Lashlee [8:94]. These results are discussed in Section 5.2.2.

For simplicity, the Q and R matrices are assumed diagonal. The Q and R matrices could be full matrices, but only the diagonal elements are used for this thesis research. Further study would involve using full Q and R matrices for potentially better tuning. The R matrix for the filter (R_f) is different from the R matrix for the truth model (R_t). Likewise, the Q matrix for the filter model (Q_f) differs from the Q matrix for the truth model (Q_t). Because the truth model has more states (24 states) than the filter model (6 states), the unmodeled states will influence the filter performance. To compensate for this, the filter is "informed" about the "real world" (the truth model) by making R_f larger than R_t. For this research, the value of R_t is set at the value found by Lashlee [8:94]. The value of R_f is modified by adding the effects of the nine unmodeled modes from the truth model in a manner to be described in the next paragraph. The values of Q_f and Q_t are also set at the value

found by Lashlee [8:94]. From these base R_f, R_t, and Q_t values, the noise level determination study is performed by varying the values of R_f and Q_f. For this study, the value of Q_f is equal to the value of Q_t; however, R_f does differ from R_t.

Since the filter model is affected by the unmodeled truth model states, the filter should be "told" of the presence of these states by increasing the value of R_f. As a first-cut increase, R_f is increased by additional terms found from the 24-state matrix product $H_tP_t(t_i^-)H_t^T$ that do not appear in $H_fP_f(t_i^-)H_f^T$. Here, H_t and $P_t(t_i^-)$ are the truth model measurement matrix and covariance matrix respectively, and H_f and $P_f(t_i^-)$ are the filter model measurement matrix and covariance matrix respectively. These additional terms are found by subtracting the 6-state matrix product $H_fP_f(t_i^-)H_f^T$ from the 24-state matrix product $H_tP_t(t_i^-)H_t^T$. This matrix difference results in a 6-by-6 matrix $(H_tP_t(t_i^-)H_t^T - H_fP_f(t_i^-)H_f^T)$. Because this research is limited to diagonal Q and R matrices, only the diagonal terms of this 6-by-6 matrix, given in Table 3, are then added to the diagonal terms of the old value of R_f ($R_f = R_t$) to become the new value of R_f ($R_f = R_t$ + the diagonal matrix composed of the diagonal terms of the 6-by-6 matrix). Using a full R_f matrix with the full 6-by-6 matrix difference added to it would probably give better results (this should be done for follow-on research). This new R_f is the base value of R_f used in the noise level determination study. Once this new R_f is found, the entire 6-by-

6 R_f matrix is scaled by scalars (as demonstrated in Table 4 of Section 5.2.2) to provide the different R_f matrices used in the evaluation.

Table 3

Diagonal Terms of the Matrix Difference

$$H_t P_t(t_i^-) H_t^T - H_f P_f(t_i^-) H_f^T)$$

Diagonal term 1,1	is	-6.624×10^{-8}
Diagonal term 2,2	is	1.048×10^{-6}
Diagonal term 3,3	is	4.458
Diagonal term 4,4	is	-0.34
Diagonal term 5,5	is	0.0
Diagonal term 6,6	is	9.77×10^{-15}

4.4.3 Robustness Analysis

While the noise level determination study is being performed, the effects of the unmodeled states will be investigated. This analysis entails performing studies on the three cases described in Section 1.4. First, a single 6-state non-adaptive filter (based on a nominal parameter value) is tested against the 24-state truth model. Second, a single 6-state artificially informed non-adaptive filter (based on the true parameter value) is tested against the 24-state truth model. Third, the moving-bank adaptive 6-state filter is tested against the 24-state truth model. The results found from this robustness analysis are compared directly to the results Lashlee found using a 6-state filter and a 6-state truth model [8:115]. In each case, the primary concern is to discover what effect the unmodeled states have on the adaptation process. The effects of

the unmodeled states will be discerned by analyzing the ability of the moving-bank MMAE to provide state estimates properly and by analyzing the relationship between state estimation and parameter estimation.

First, the 6-state non-adaptive filter study is performed by starting the filter at a parameter which is not the true parameter. The true parameter is at (7,6). The parameter used to start the filter is at (5,5), in the center of the parameter space. The probabilities of the other 8 filters in the bank are forced to zero to keep them from affecting the state estimates. The results are presented in Section 5.2.3.

Second, the artificially informed non-adaptive 6-state filter study is performed. For this study, a single filter is artificially "told" the true parameters. The probabilities of the other 8 filters in the bank are forced to zero to keep them from affecting the state estimates. The filter is based on the true parameter (7,6). The results of this study are given in Section 5.2.3.

Finally, the 6-state filter and 24-state truth model study is performed by simply evaluating the results provided by the noise determination study. The goal of this study (and the thesis research) is to determine whether or not (and to what extent) the unmodeled states affect the adaptation process. This

study will also be accomplished using two cases: the first case entails initially centering the bank at the true parameter and the second case entails initially centering the bank at a parameter which is not the true parameter. The effects of the unmodeled states are readily visible by the ability of the 6-state filter to provide accurate estimates. The results of this study are given in Section 5.2.3.

4.4.4 Controller Study

Once the robustness study is complete, a controller study is performed. In this study, the performance of the moving-bank MMAC will be evaluated. Figure 2 depicts MMAC algorithm. The state estimates produced by each elemental Kalman filter are fed into the controller gains, $G_c^*(\underline{a}_i)$, associated with that particular elemental filter (refer to Equation (43)). These individual control outputs (one for each elemental Kalman filter in the bank) are then probabilistically weighted and summed (refer to Equation (42)) to provide the control input. The primary goal of the controller study is to determine the effects of unmodeled states on the ability of the MMAC to provide adequate control. The effects of the unmodeled states on the control should be analyzed by determining how well the controller regulates the states to zero, as compared to corresponding results against a 6-state truth model. However, since the sample statistics of the states are not extracted, the mean plus or

minus the standard deviation of the state estimation errors is used to measure the "stability" of the controller.

The controller study is performed by starting the bank at either the true parameter or at some parameter that is not the true parameter. The Q_t and R_t matrices are set at the values found by Lashlee [8:94]. The matrix R_f is initialized at the same value used in Section 4.4.2. As in the noise level determination study, the R_f matrix differs from the R_t matrix. Unlike the noise level determination study, the Q_f matrix differs from the Q_t matrix. Allowing the R_f and Q_f matrices to vary from Q_t and R_t provides for a true R and Q tuning. This true tuning is incorporated because (as explained in the next chapter) allowing only for R_f tuning (as done in the previous studies) is not adequate for stable and accurate control. This study is then performed in the same manner as the noise level determination study--by varying the Q_f and R_f values by scalar multiples to produce accurate state estimates. The results of this study are given in Section 5.2.4.

4.5 Summary

Section 4.2 gives the reasons for and describes the Monte Carlo analysis technique. Section 4.3 describes the software used in the simulation, and the final section describes the detailed simulation plan. The next chapter gives the results of the simulation and a discussion of important trends, and the

final chapter gives the conclusions and recommendations for further research.

V. Results

5.1 Introduction

The Monte Carlo simulation results are presented in this chapter. The goal of this thesis effort is to investigate the robustness of the MMAE/MMAC algorithm in the presence of unmodeled truth model states. First, computer programs are transported from the CDC Cyber to the Elxsi 6400 superminicomputer. Second, the code used by Lashlee [8:70-88] is modified to provide a 24-state truth model along with the 6-state filter model. Third, the modified code is run in the Monte Carlo simulations described in Section 4.4.

5.2 Software Simulation

The software simulation involves four steps. The first step is to duplicate Lashlee's results by running the unmodified and modified software on the Elxsi. The second step is to perform the noise level determination study. The third step is the robustness analysis, and the fourth step is the controller study. Note that (except where noted) throughout these studies, the bank is initially centered on the true parameter. It will be shown later that the moving bank is unable to provide accurate estimates when initially centered at a parameter that is not the true parameter. All plots shown in this chapter are plots of the mean plus or minus the standard deviation (σ) of the state estimate error ($T\underline{x}_t(t_i^+) - \hat{\underline{x}}(t_i^+)$). This state estimate error is

taken at time t_i^+, which is right after the state estimate update. Note that for this thesis, the plots given for the MMAC performance are also mean plus or minus the standard deviation of the estimation error rather than the total true state values. This is not the best representation of controller performance, but these plots were the only representation available.

5.2.1 Duplication of Past Work

The duplication effort is conducted as described in Section 4.4.1 and the results are presented in this section. Figure 11 is a series of plots for Lashlee's original unmodified software performed on the Elxsi. Figure 12 is a series of like plots of the modified (but with states 7-24 set to zero) software performed on the Elxsi. As seen by the plots of Figures 11 and 12, the modified programs provide a simulation performance comparable to the original unmodified programs and Lashlee's actual thesis results [8:95]. The slight differences in the plots can be accounted for by the numerical inaccuracies incurred when calculating the 24-state transition matrix versus the 6-state transition matrix and machine precision between the Elxsi and Cyber. The goal of this study is to achieve results that are similar for each case but not necessarily exact.

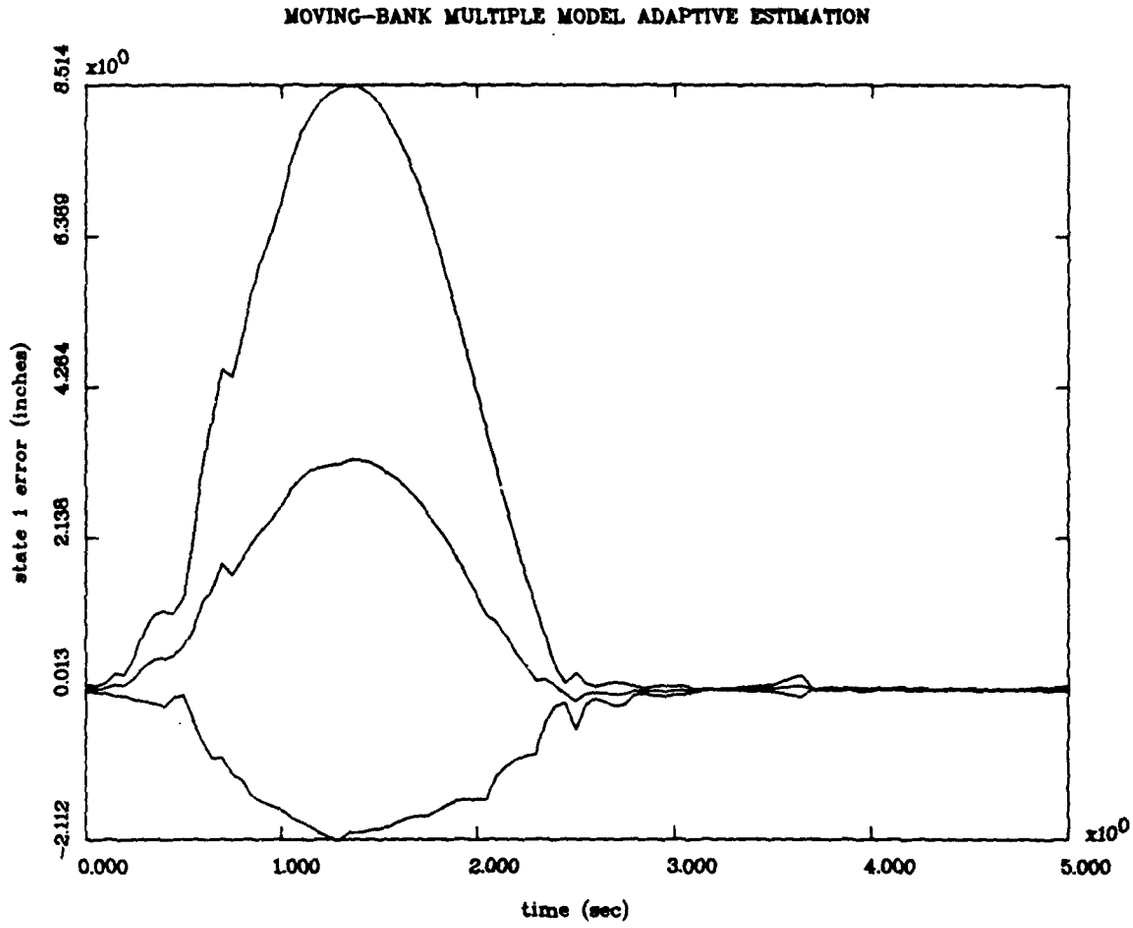

Figure 11(a) State 1
State Estimation Errors
Mean ± 1σ
Unmodified Software Run on Elxsi
to duplicate Lashlee's results

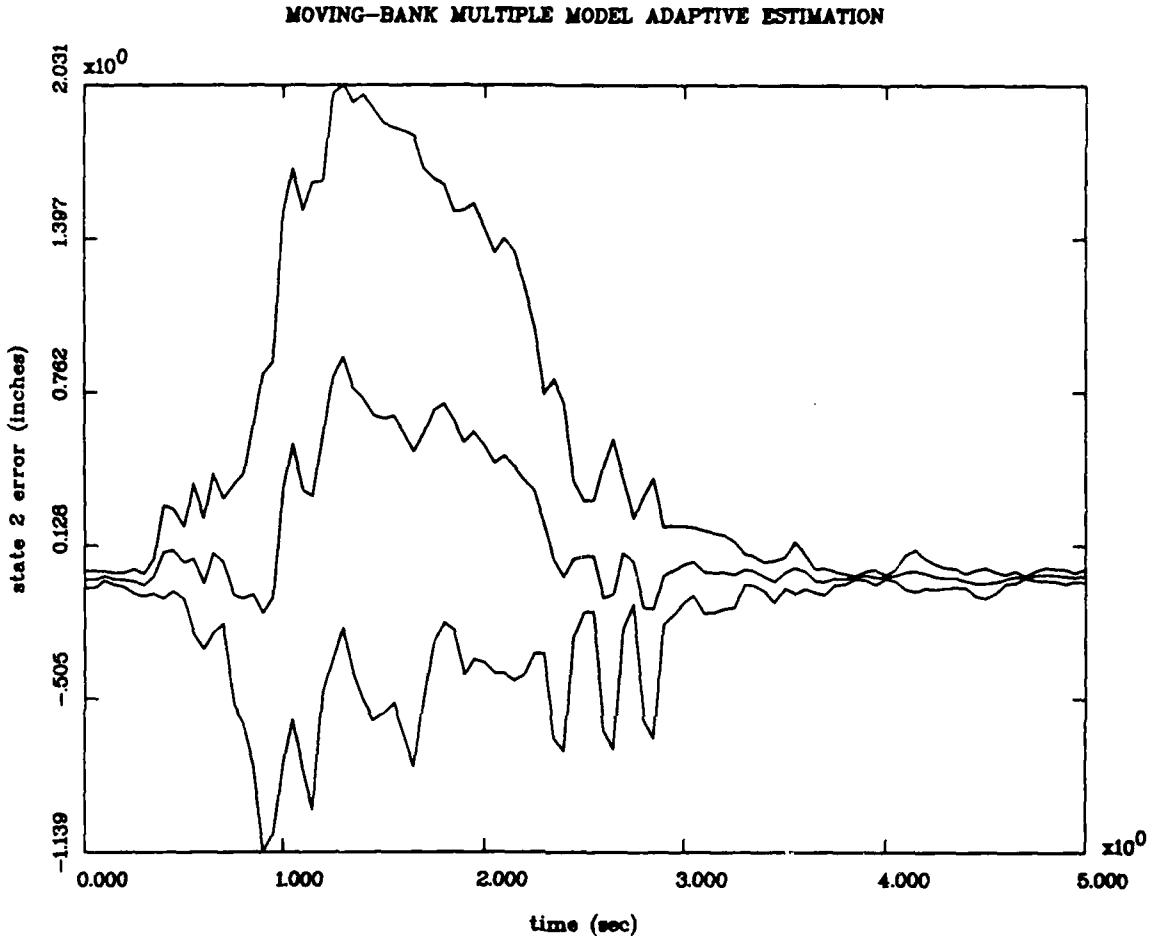

Figure 11(b) State 2
State Estimation Errors
Mean ± 1σ
Unmodified Software Run on Elxsi
to duplicate Lashlee's results

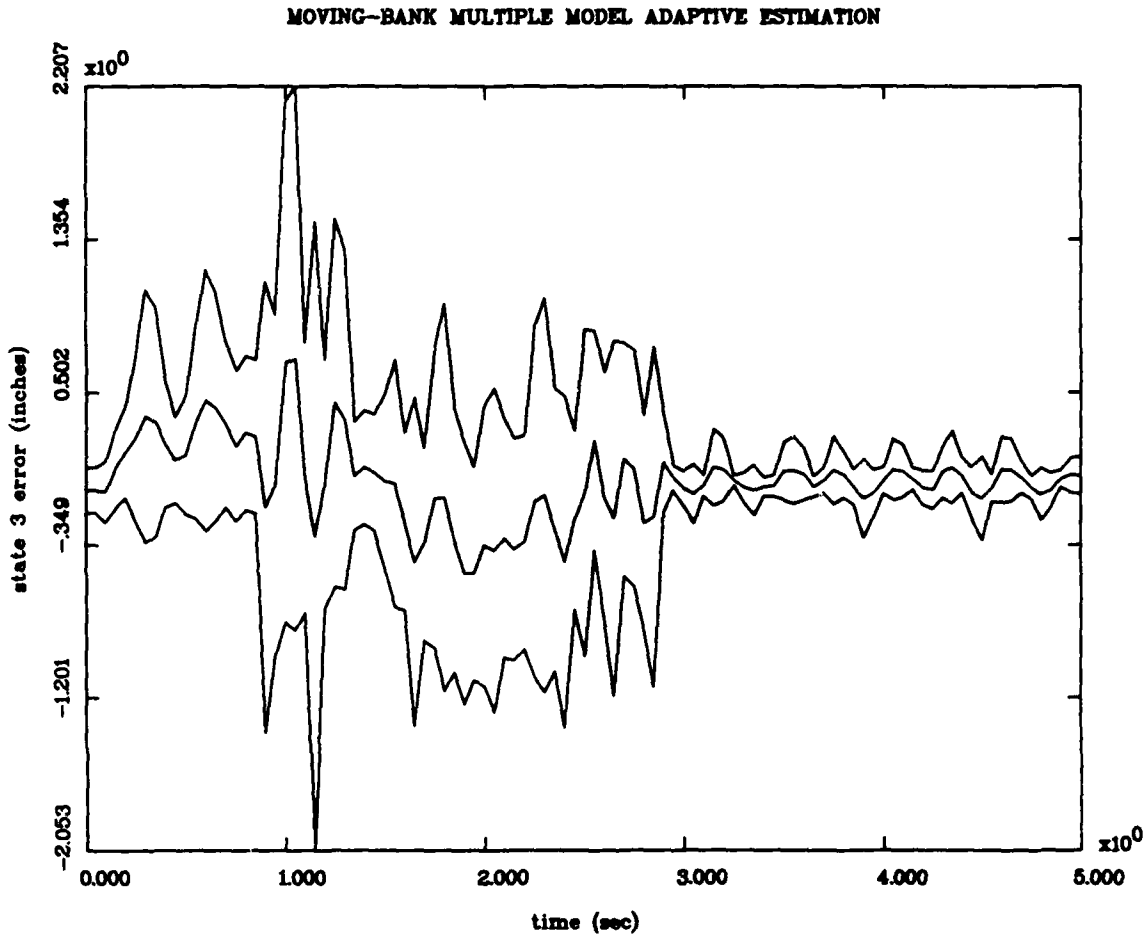

Figure 11(c) State 3
State Estimation Errors
Mean ± 1σ
Unmodified Software Run on Elxsi
to duplicate Lashlee's results

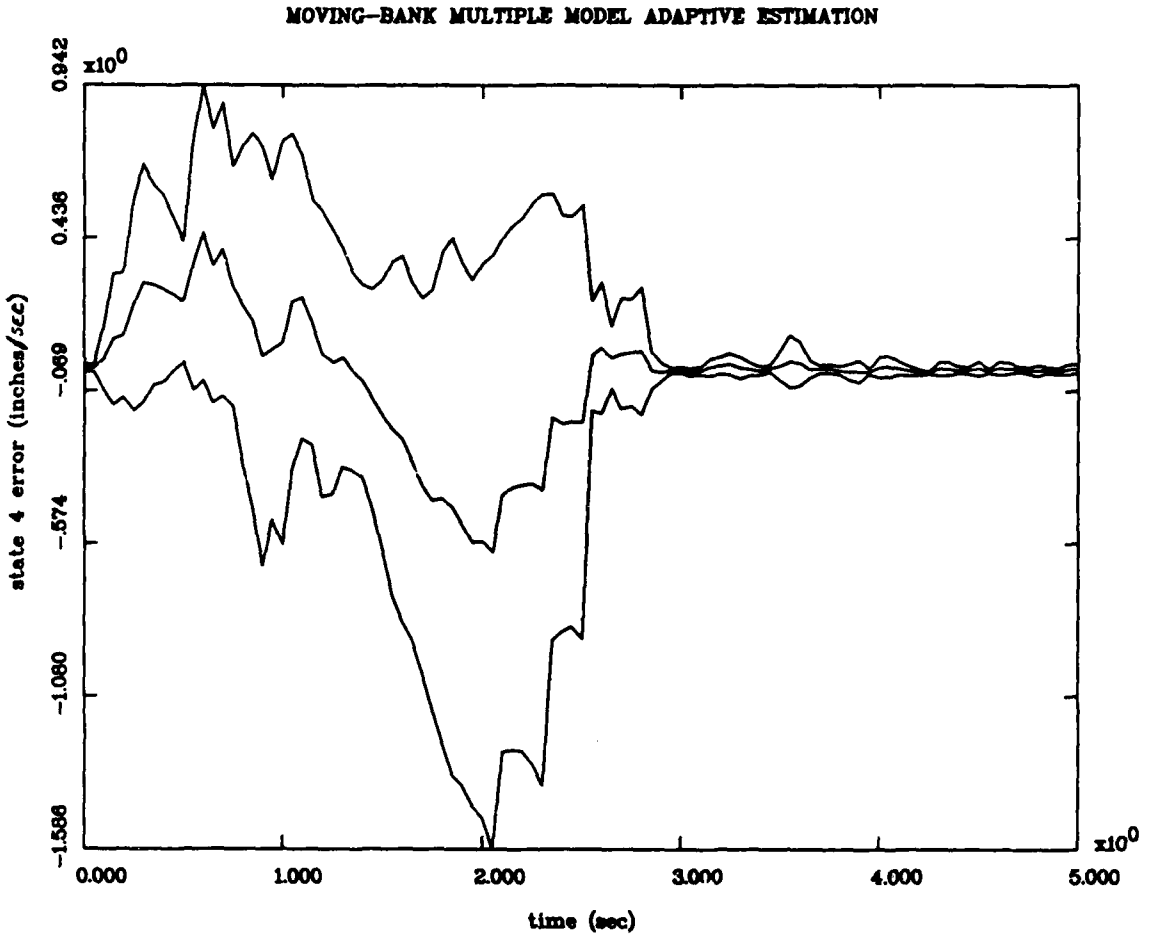

Figure 11(d) State 4
State Estimation Errors
Mean ± 1σ
Unmodified Software Run on Elxsi
to duplicate Lashlee's results

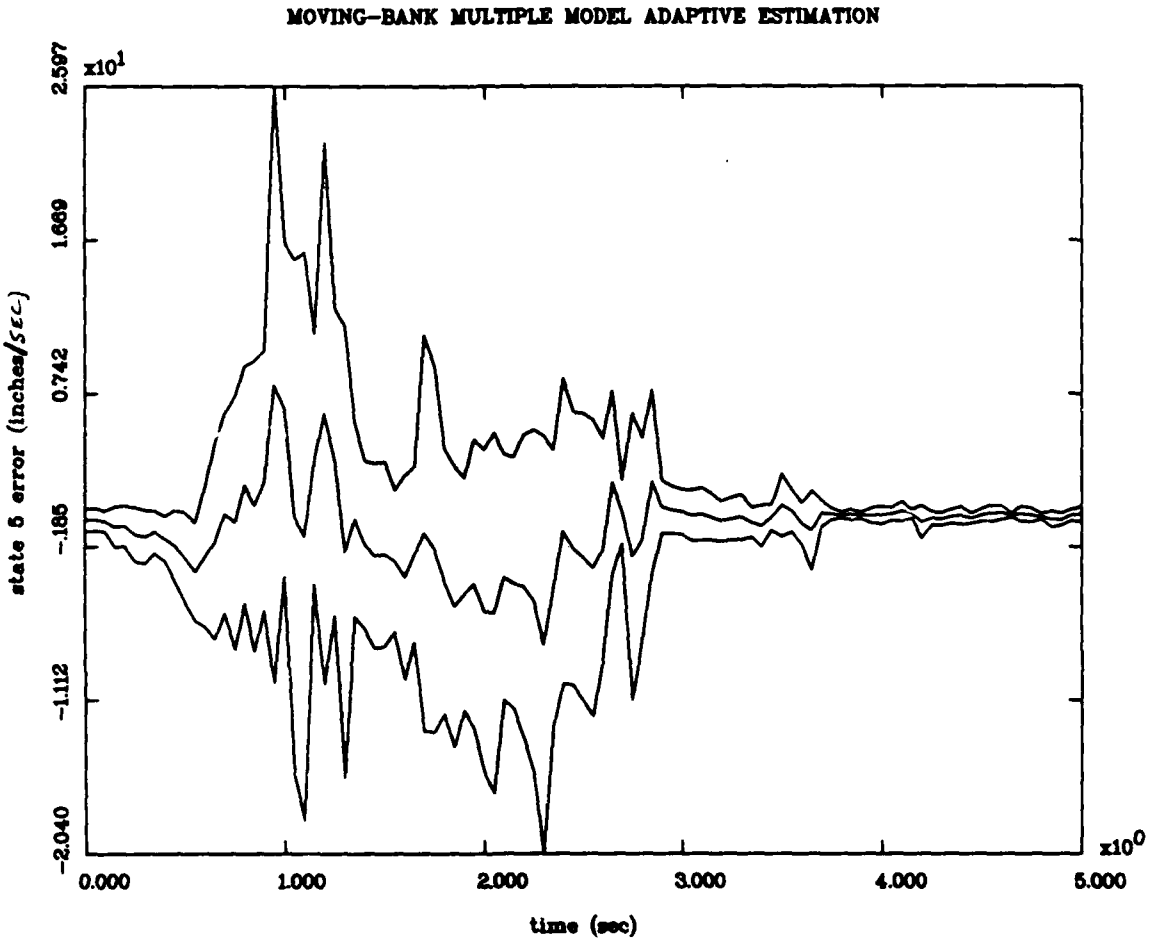

Figure 11(e) State 5
State Estimation Errors
Mean ± 1σ
Unmodified Software Run on Elxsi
to duplicate Lashlee's results

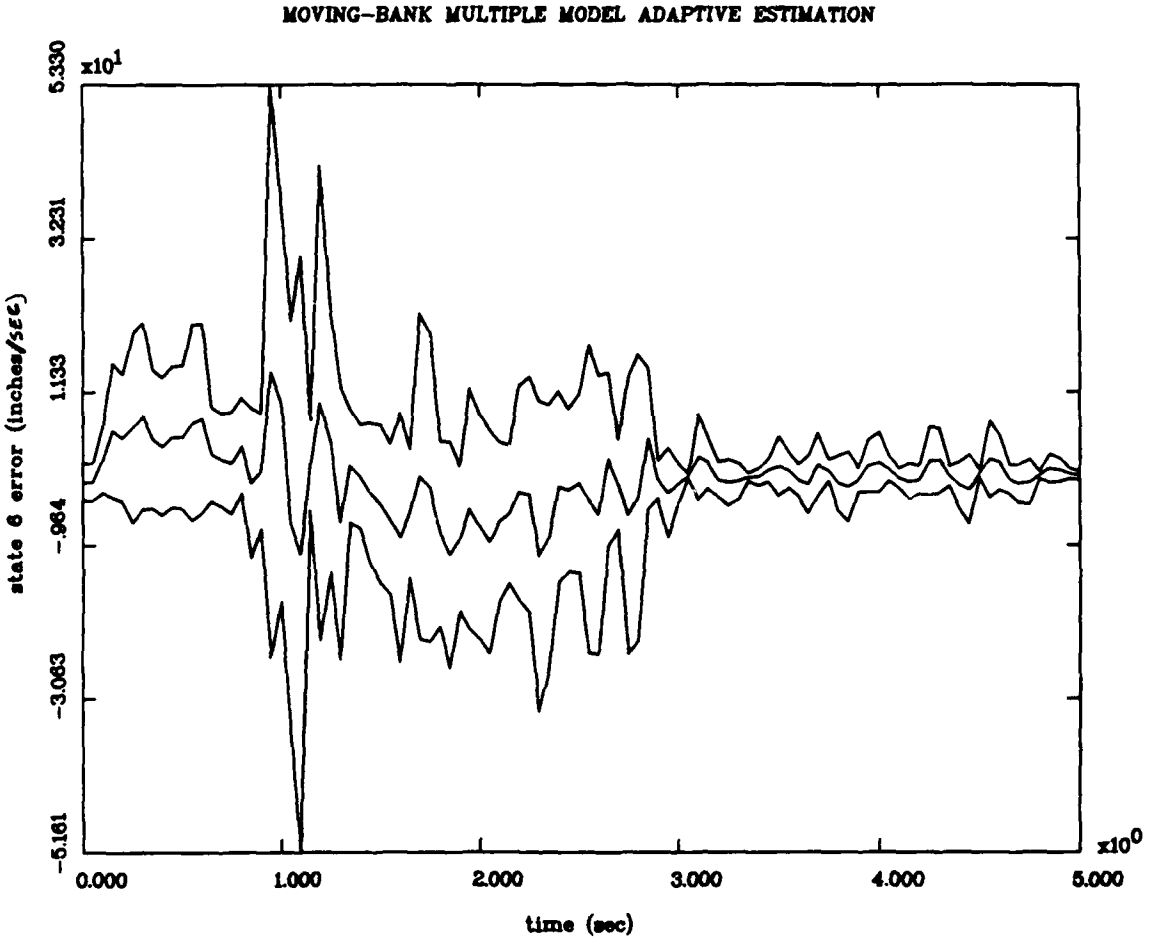

Figure 11(f) State 6
State Estimation Errors
Mean ± 1σ
Unmodified Software Run on Elxsi
to duplicate Lashlee's results

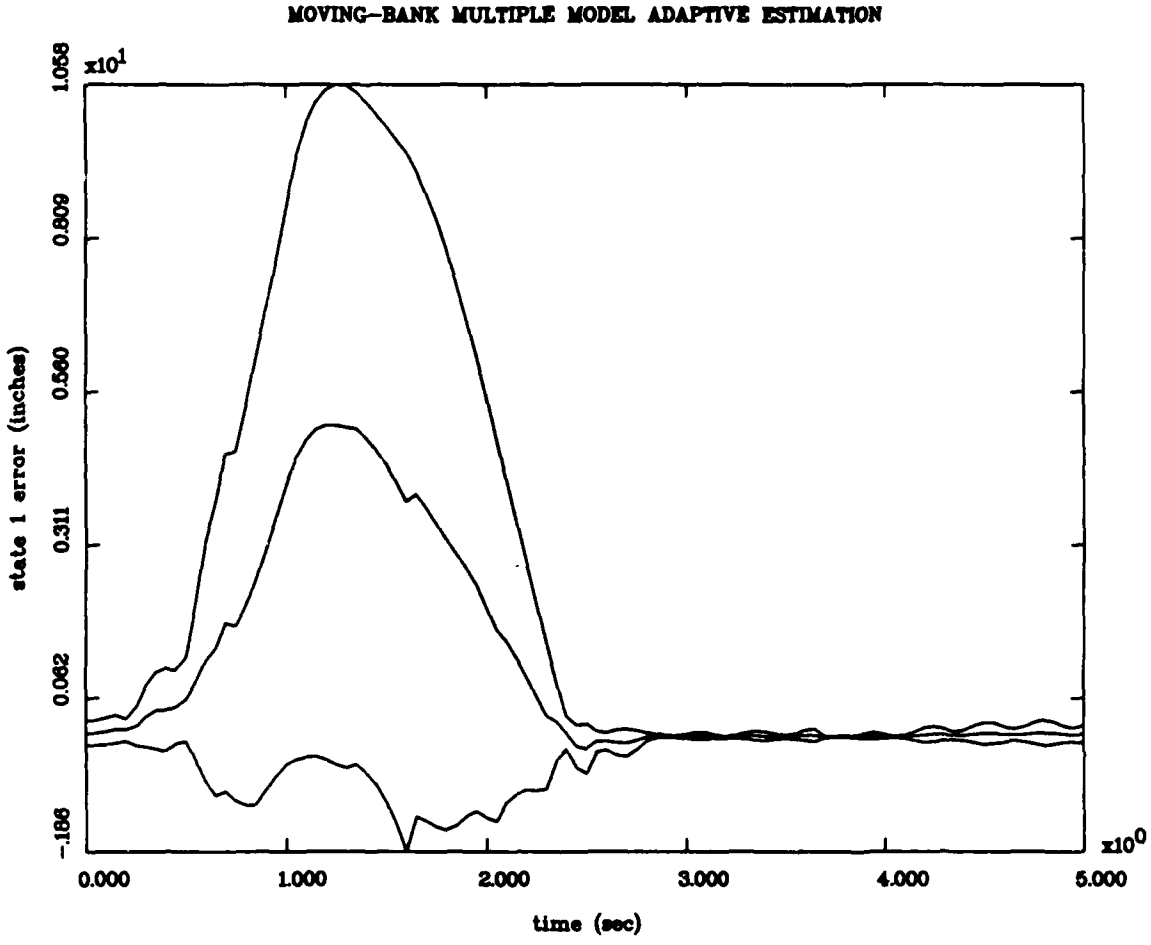

Figure 12(a) State 1
State Estimation Errors
Mean ± 1σ
Modified Software Run on Elxsi
to duplicate Lashlee's results

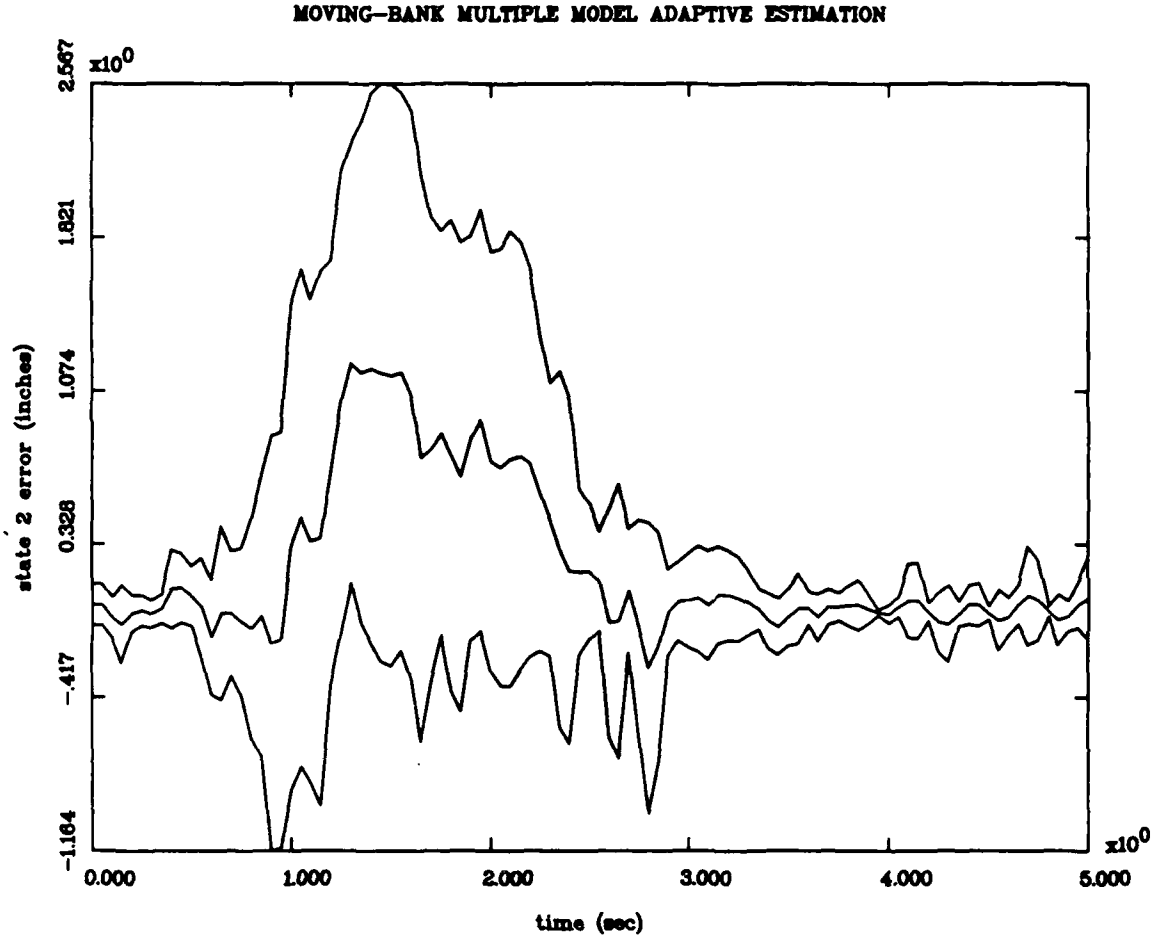

Figure 12(b) State 2
State Estimation Errors
Mean ± 1σ
Modified Software Run on Elxsi
to duplicate Lashlee's results

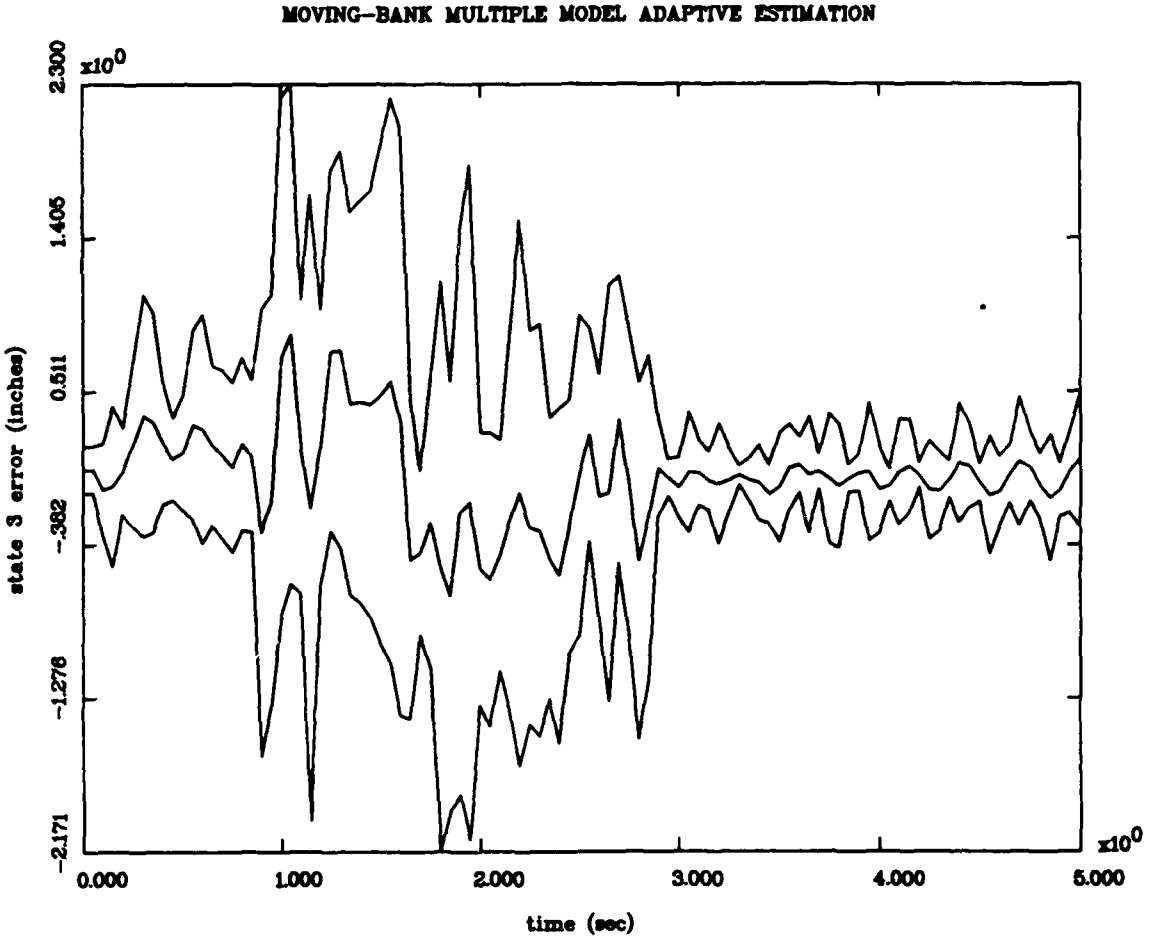

Figure 12(c) State 3
State Estimation Errors
Mean ± 1σ
Modified Software Run on Elxsi
to duplicate Lashlee's results

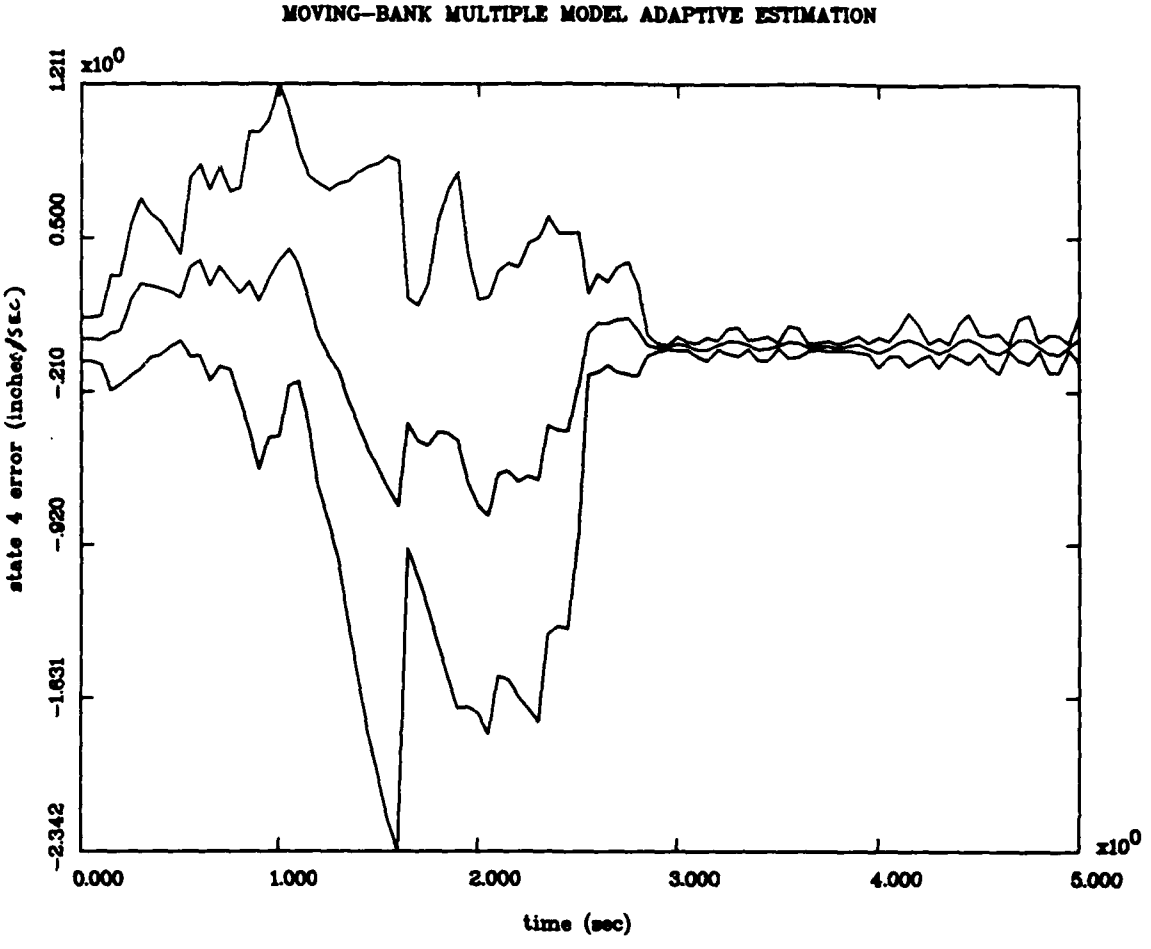

Figure 12(d) State 4
State Estimation Errors
Mean ± 1σ
Modified Software Run on Elxsi
to duplicate Lashlee's results

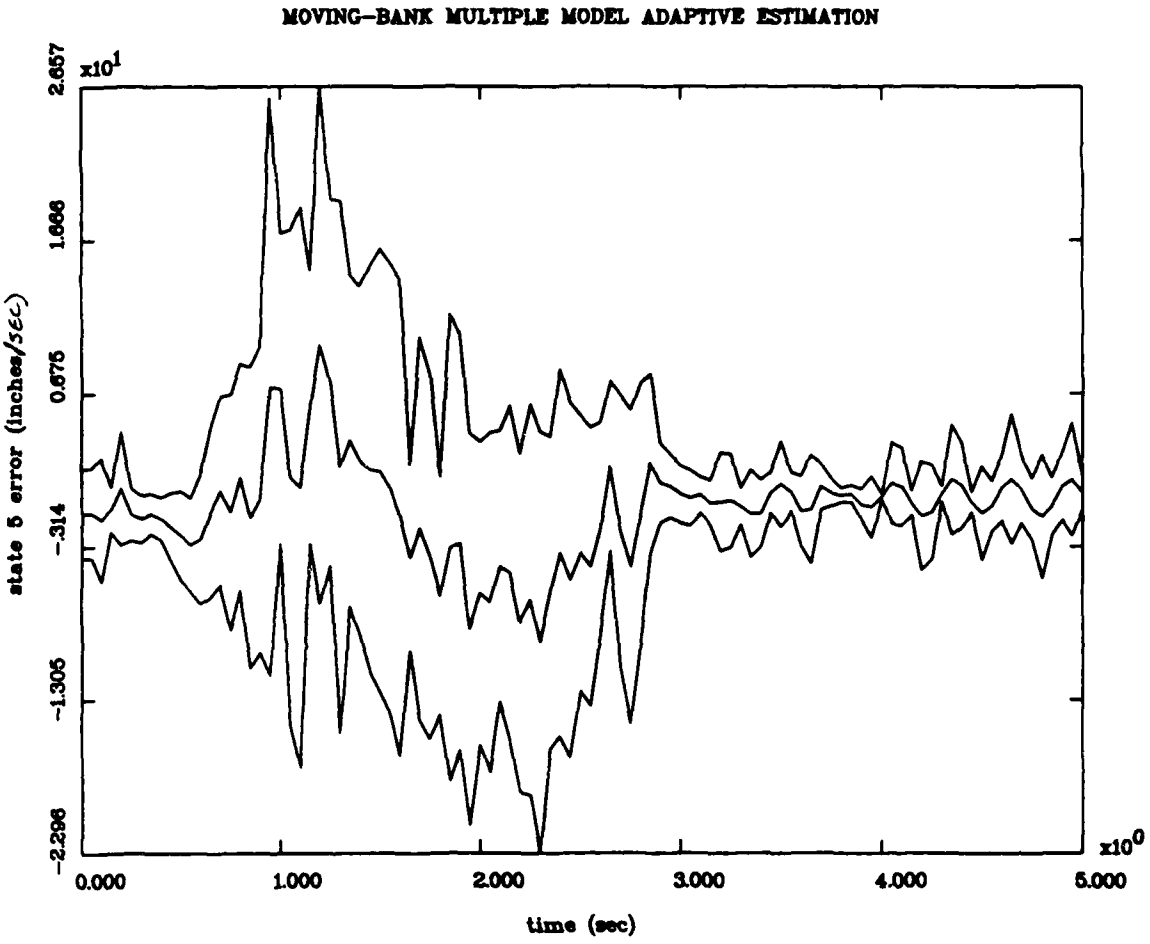

Figure 12(e) State 5
State Estimation Errors
Mean ± 1σ
Modified Software Run on Elxsi
to duplicate Lashlee's results

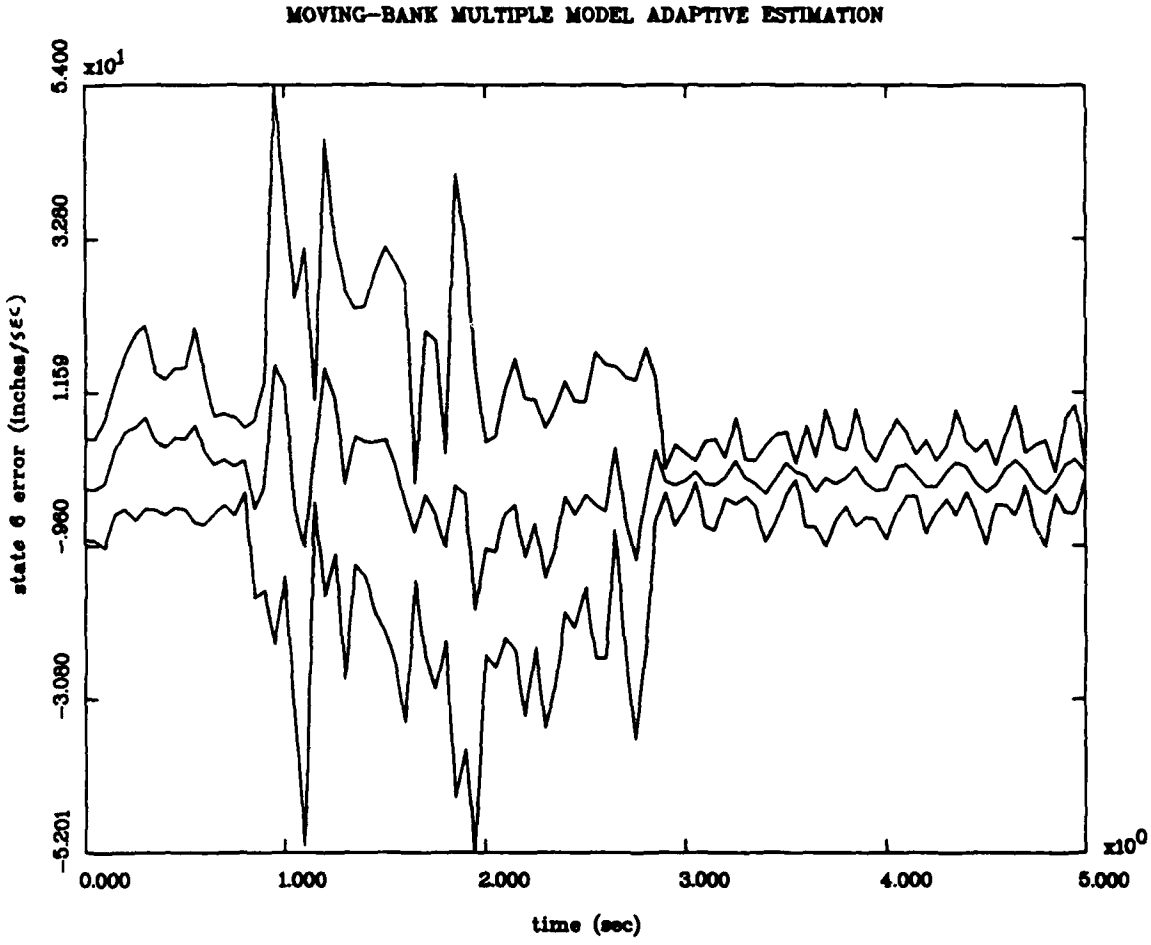

Figure 12(f) State 6
State Estimation Errors
Mean ± 1σ
Modified Software Run on Elxsi
to duplicate Lashlee's results

5.2.2 Noise Level Determination

As mentioned in Section 4.4.2, the second step of the simulation is to find choices for the measurement noise covariance matrix (R) and the dynamics noise strength matrix (Q) which provide accurate state estimates. During this study, the measurement noise covariance matrix for the filter model (R_f) is set to a different value than the measurement noise covariance matrix for the truth model (R_t). However, the dynamics noise covariance matrix for the truth and filter models (Q_t and Q_f) are equal to each other.

Because Lashlee [8:198] found that the value of R plays a very important role in the performance of the moving-bank algorithm, and because the major difference between the truth and reduced order model representations is the inclusion or deletion of the higher order modes in the measurement relationships, it was originally planned to tune only the R_f matrix and not the Q_f matrix. As it turns out, it would have been better to tune both the R_f and Q_f matrices. Because of time constraints, this R_f and Q_f tuning was not possible for the noise level determination study; however, R_f and Q_f are tuned for the controller study as described in Section 5.2.4. In fact, to achieve stable and accurate state estimates, the Q matrix had to be changed in a sensitivity analysis manner. For this study, as for all the studies done in this thesis, the matrix R_t remains at the value found by Lashlee [8:94]. Likewise, the matrices Q_t and Q_f start

at the values found by Lashlee [8:94] but can vary together by a scalar multiplication factor.

As mentioned in Section 4.4.2, the noise level determination starts with $R_t = R_f = R$ (found by Lashlee) and using the Q found by Lashlee [8:94]. The first plot shown in Figure 13 appears to be unstable, but the remaining plots of Figure 13 appear stable. This apparent instability merits further investigation, so Figure 14 shows the plots under the same conditions except that 100 Monte Carlo runs are performed rather than 10.

Figure 14 demonstrates that the moving bank provides essentially stable estimates, but tremendous peaks in the standard deviations of the state estimate errors can be seen. These large peaks may be due to the fact that the moving bank never locks on to accurate parameter estimates. However, the tremendous change in the scale represented by Figures 13 and 14 suggests something else may be wrong--there may be a phenomenon seen in Monte Carlo runs 11-100 that is not evidenced in runs 1-10. This Large swing in peak values between the two figures also suggests that there is a total loss of lock of parameter and states estimates. Therefore, as the parameter estimates provide worse state estimates, the state estimate error peaks, and as the parameter estimates provide better state estimates, the state estimate error decreases in a continuous "cycle" as shown by Figure 14.

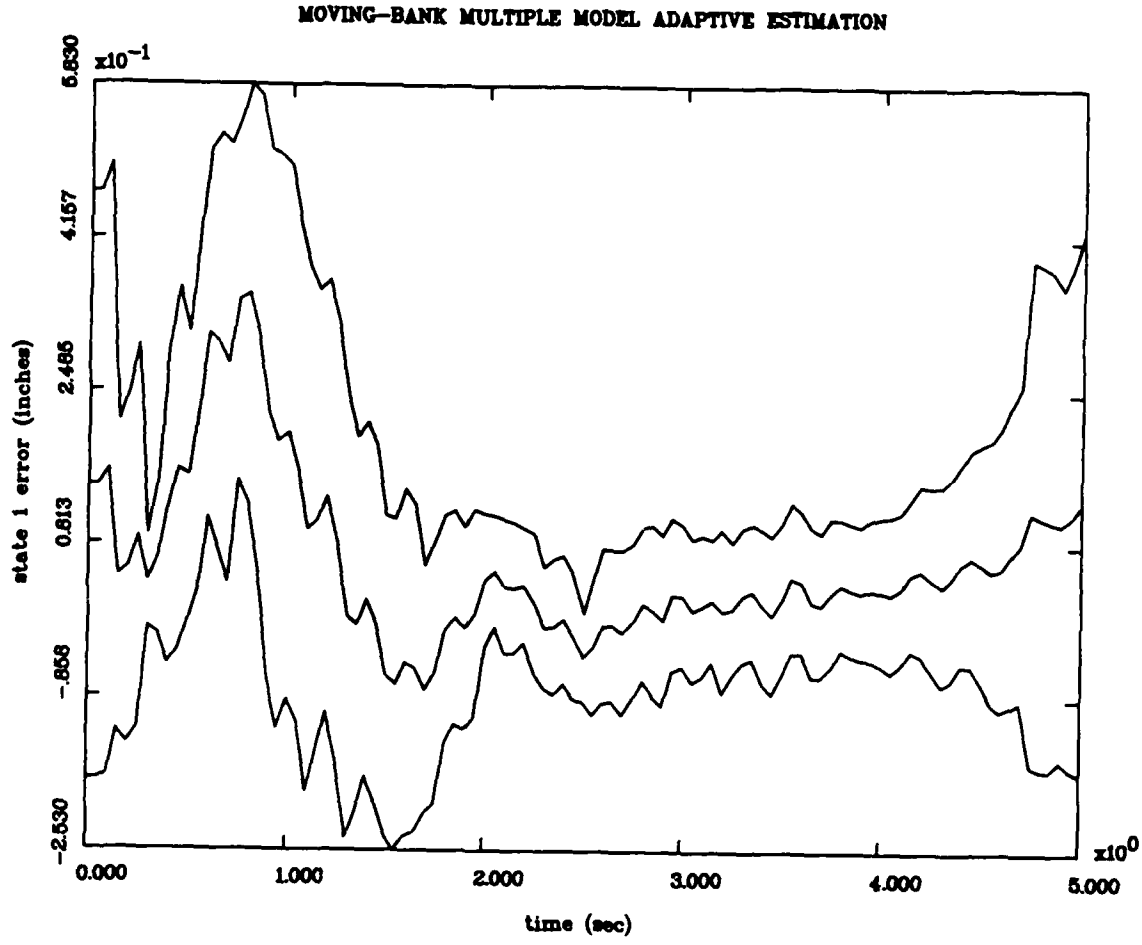

Figure 13(a) State 1
State Estimation Errors
Mean ± 1σ
Noise Level Determination Study
For R1Q1e Case
with 10 Monte Carlo Runs

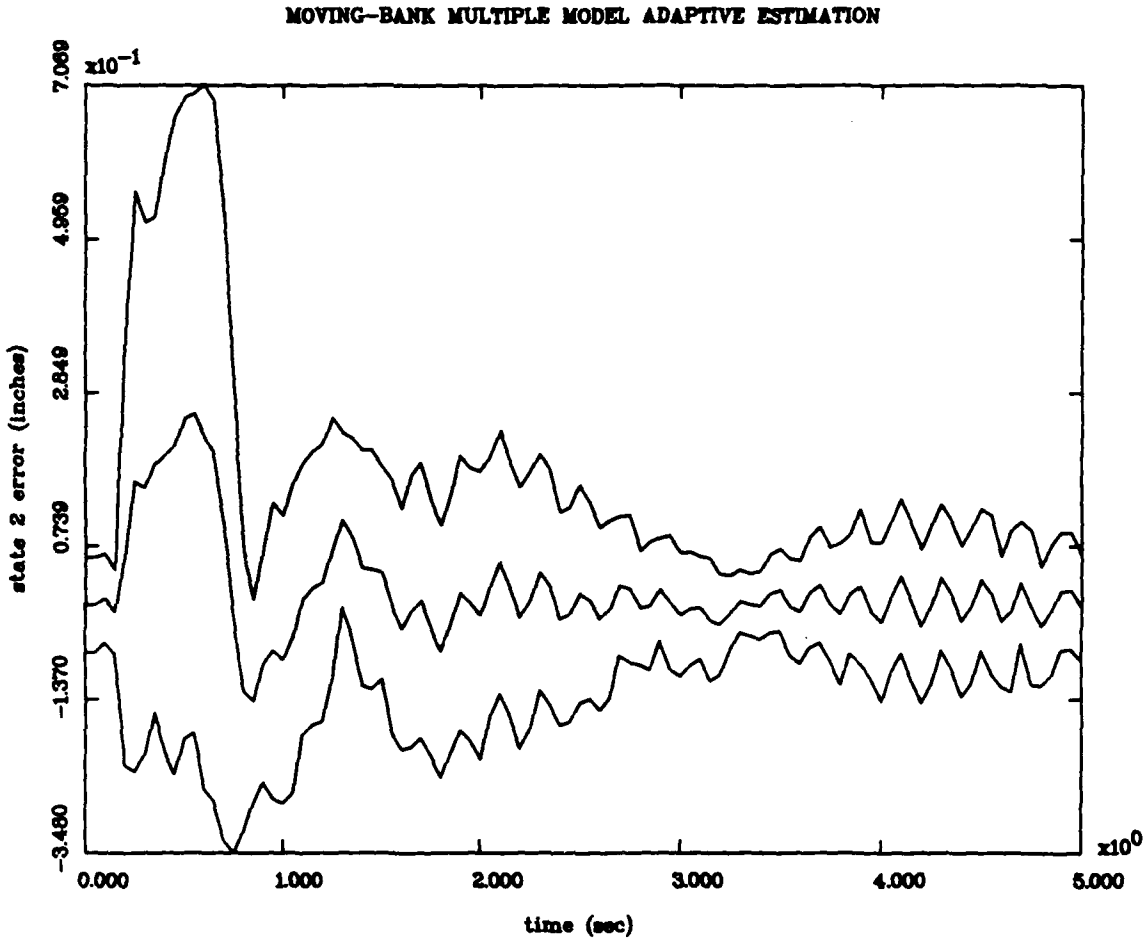

Figure 13(b) State 2
State Estimation Errors
Mean ± 1σ
Noise Level Determination Study
For R1Q1e Case
with 10 Monte Carlo Runs

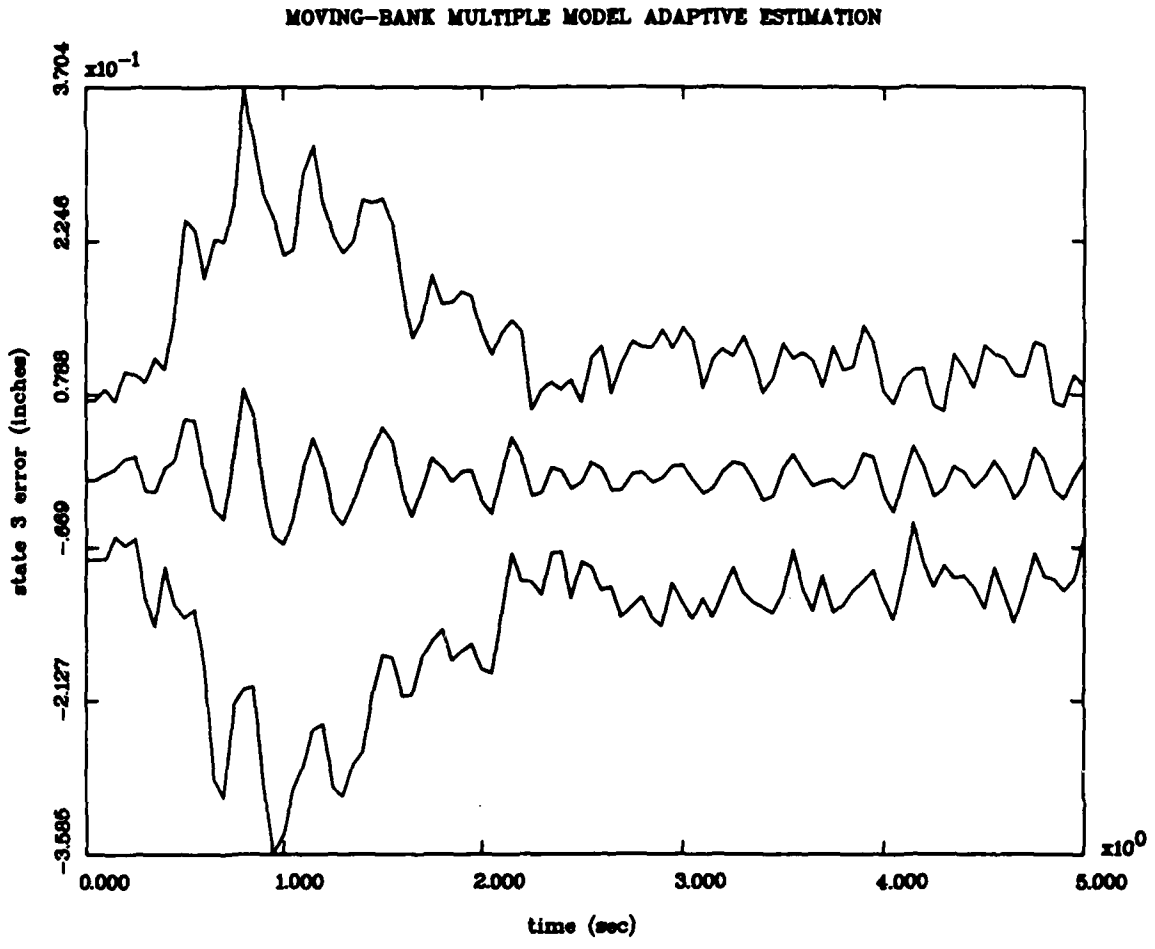

Figure 13(c) State 3
State Estimation Errors
Mean ± 1σ
Noise Level Determination Study
For R1Q1e Case
with 10 Monte Carlo Runs

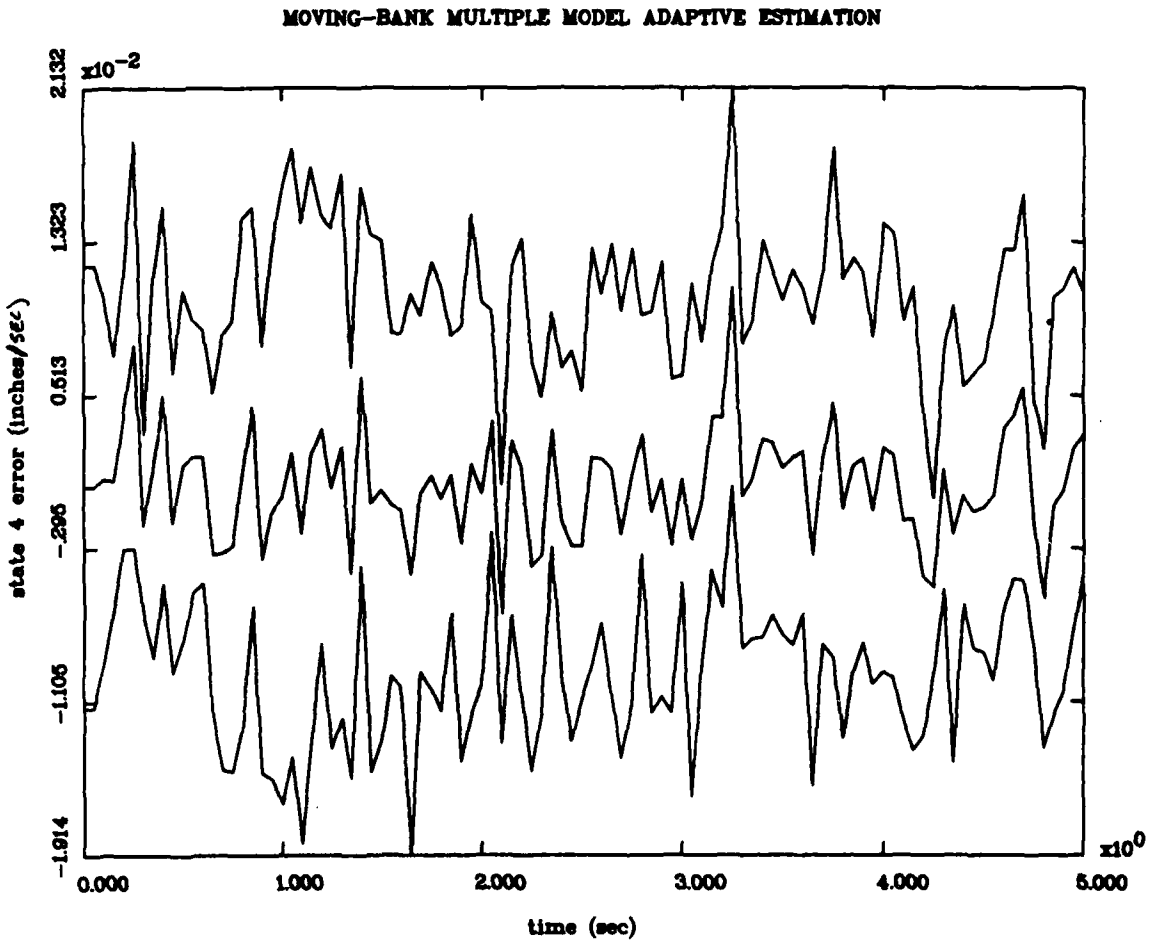

Figure 13(d) State 4
State Estimation Errors
Mean ± 1σ
Noise Level Determination Study
For R1Q1e Case
with 10 Monte Carlo Runs

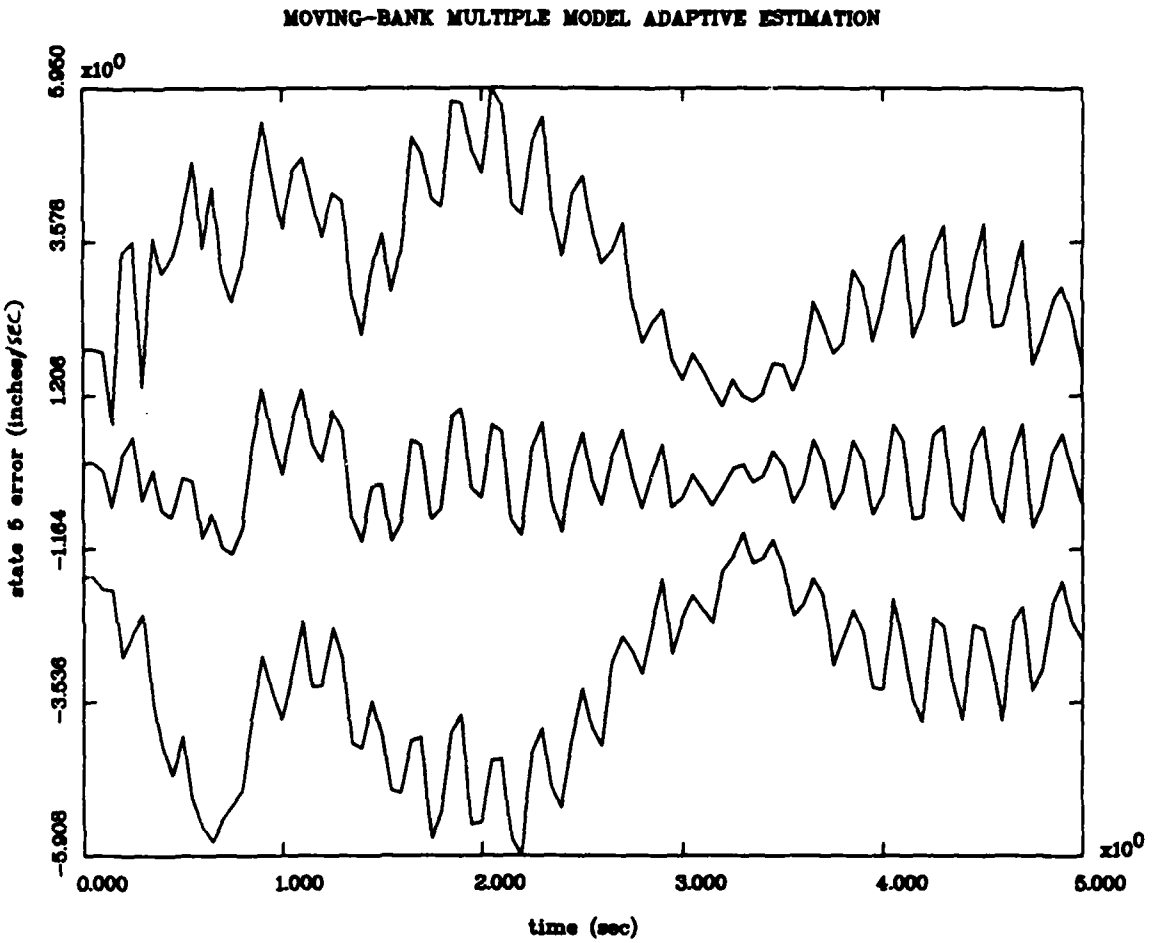

Figure 13(e) State 5
State Estimation Errors
Mean ± 1σ
Noise Level Determination Study
For R1Q1e Case
with 10 Monte Carlo Runs

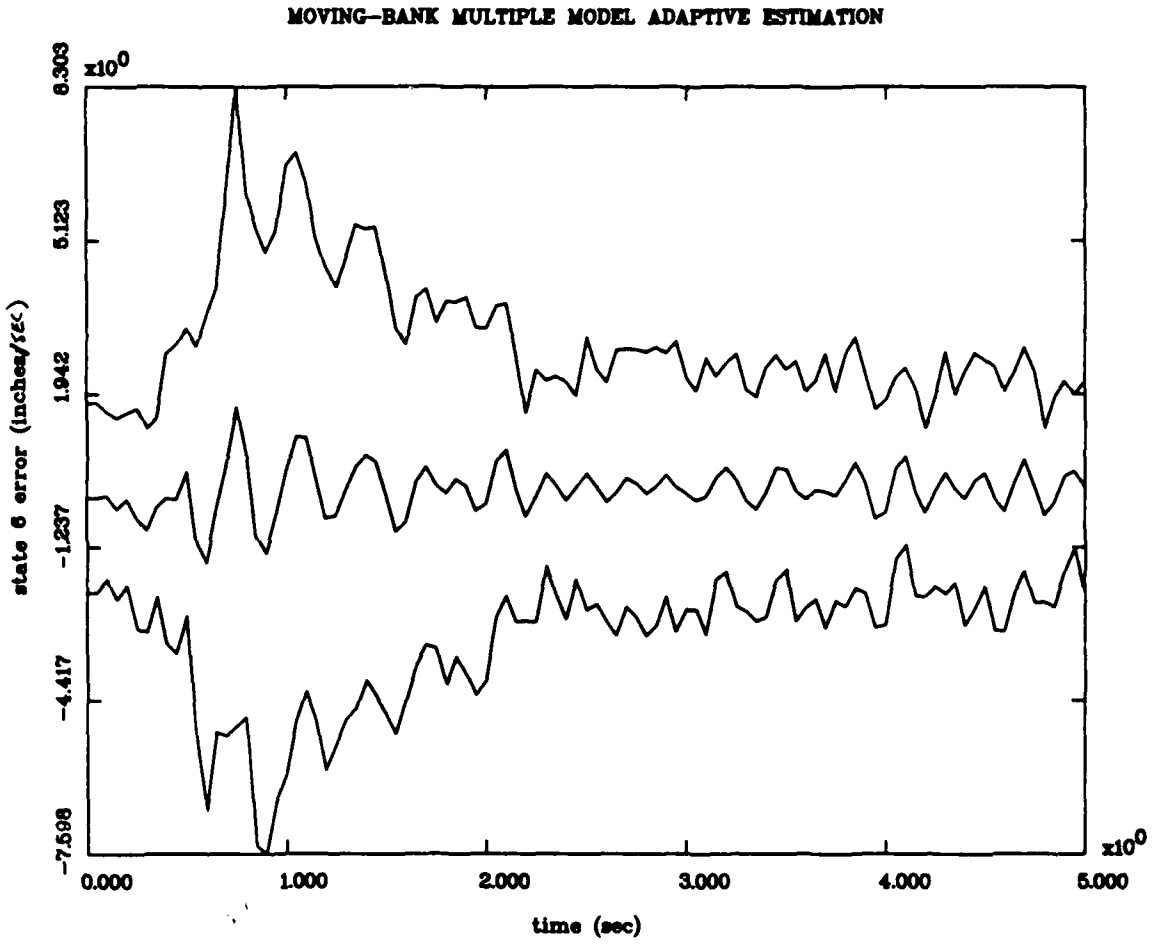

Figure 13(f) State 6
State Estimation Errors
Mean ± 1σ
Noise Level Determination Study
For R1Q1e Case
with 10 Monte Carlo Runs

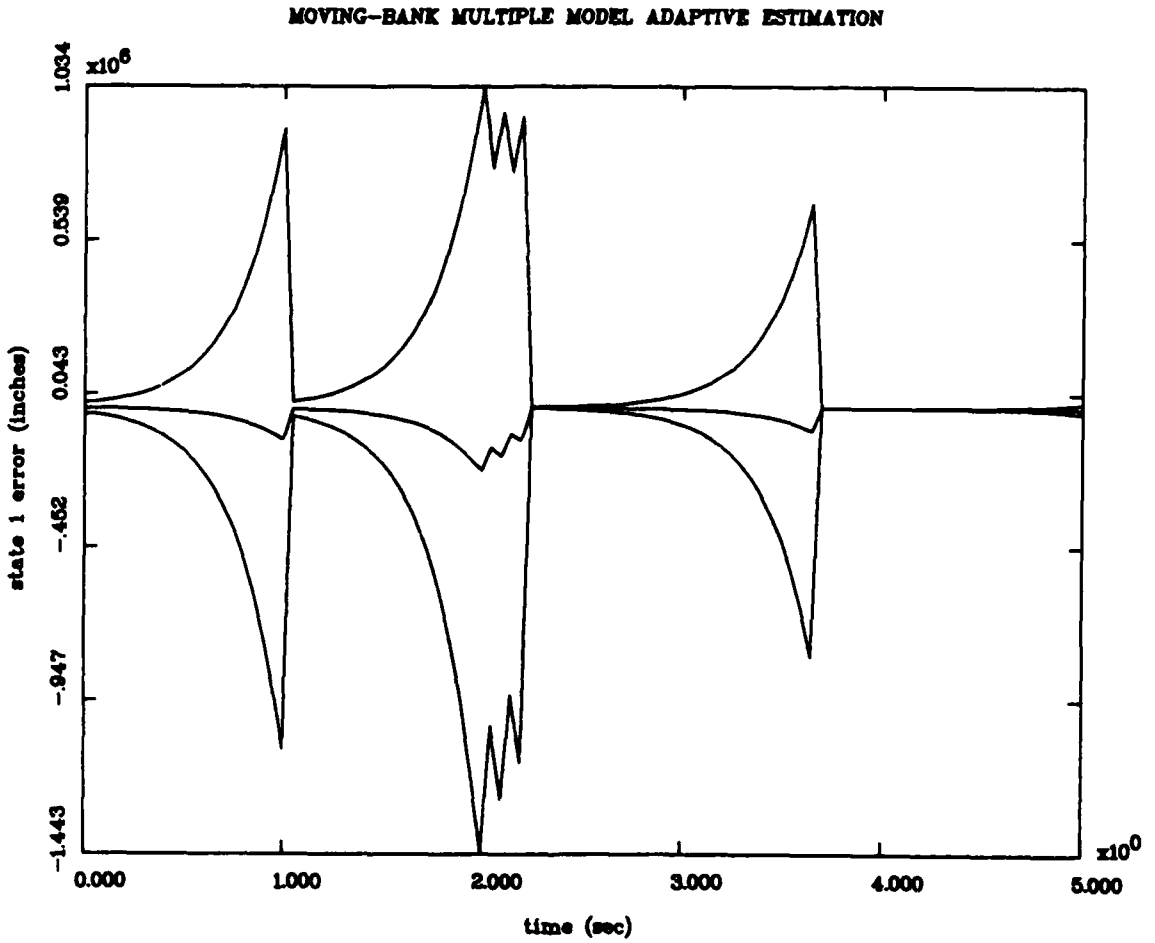

Figure 14(a) State 1
State Estimation Errors
Mean ± 1σ
Noise Level Determination Study
For R1Q1e Case
with 100 Monte Carlo Runs

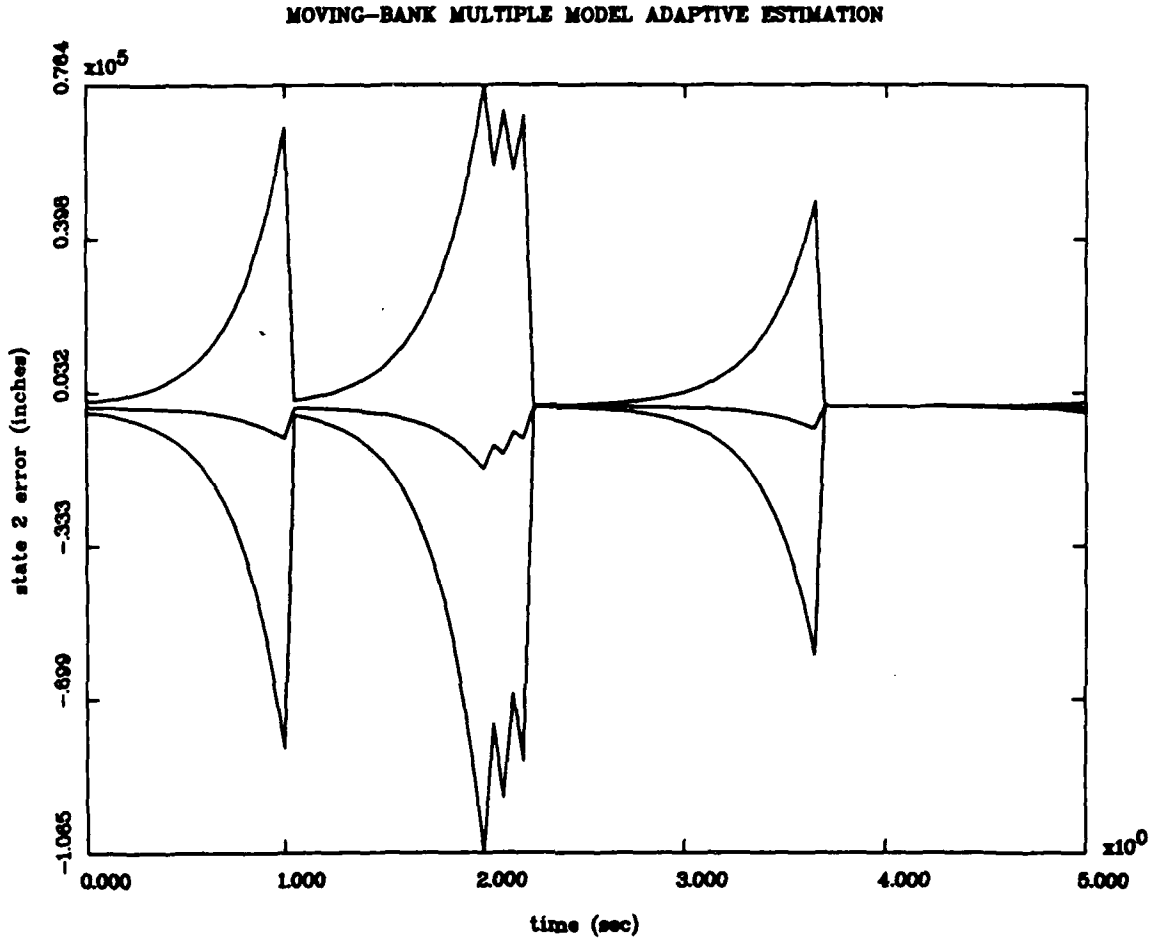

Figure 14(b) State 2
State Estimation Errors
Mean ± 1σ
Noise Level Determination Study
For R1Q1e Case
with 100 Monte Carlo Runs

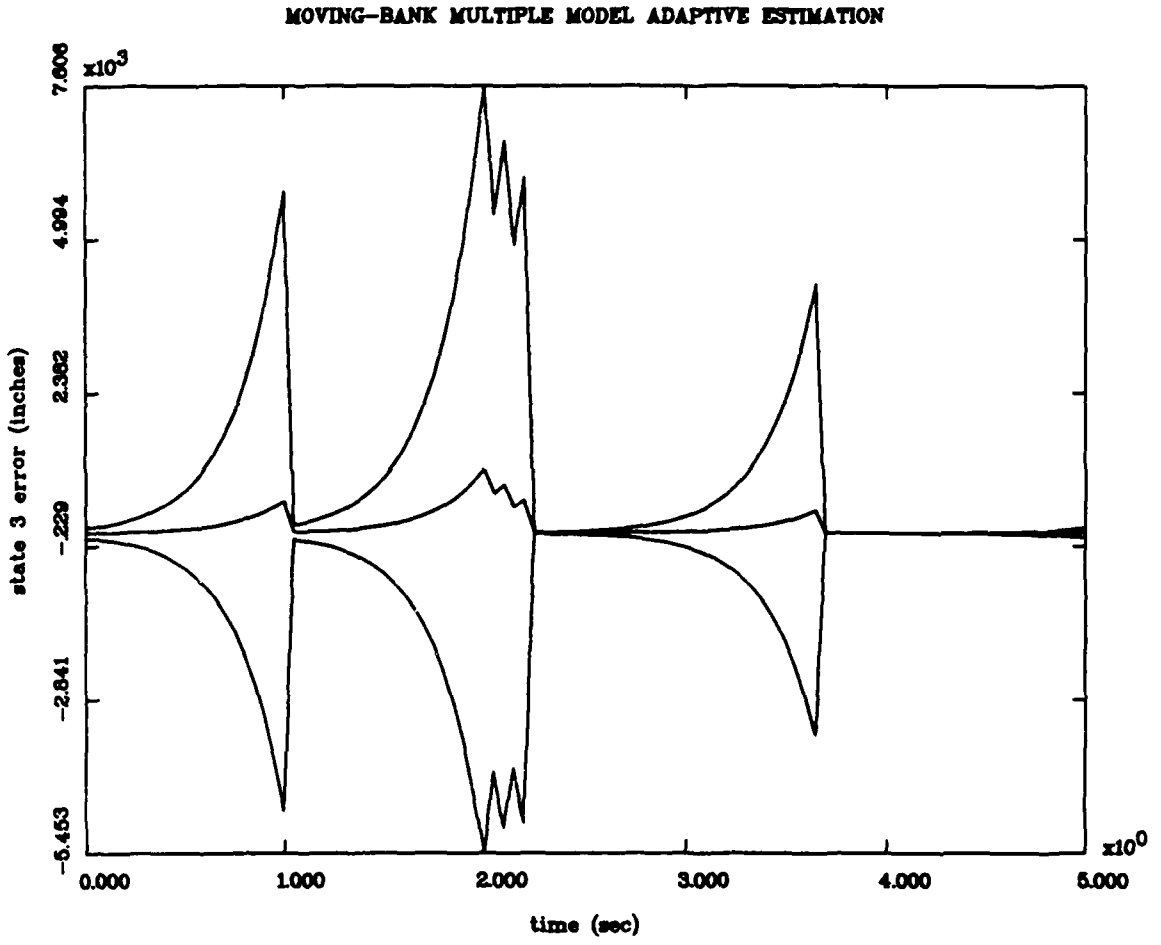

Figure 14(c) State 3
State Estimation Errors
Mean ± 1σ
Noise Level Determination Study
For R1Q1e Case
with 100 Monte Carlo Runs

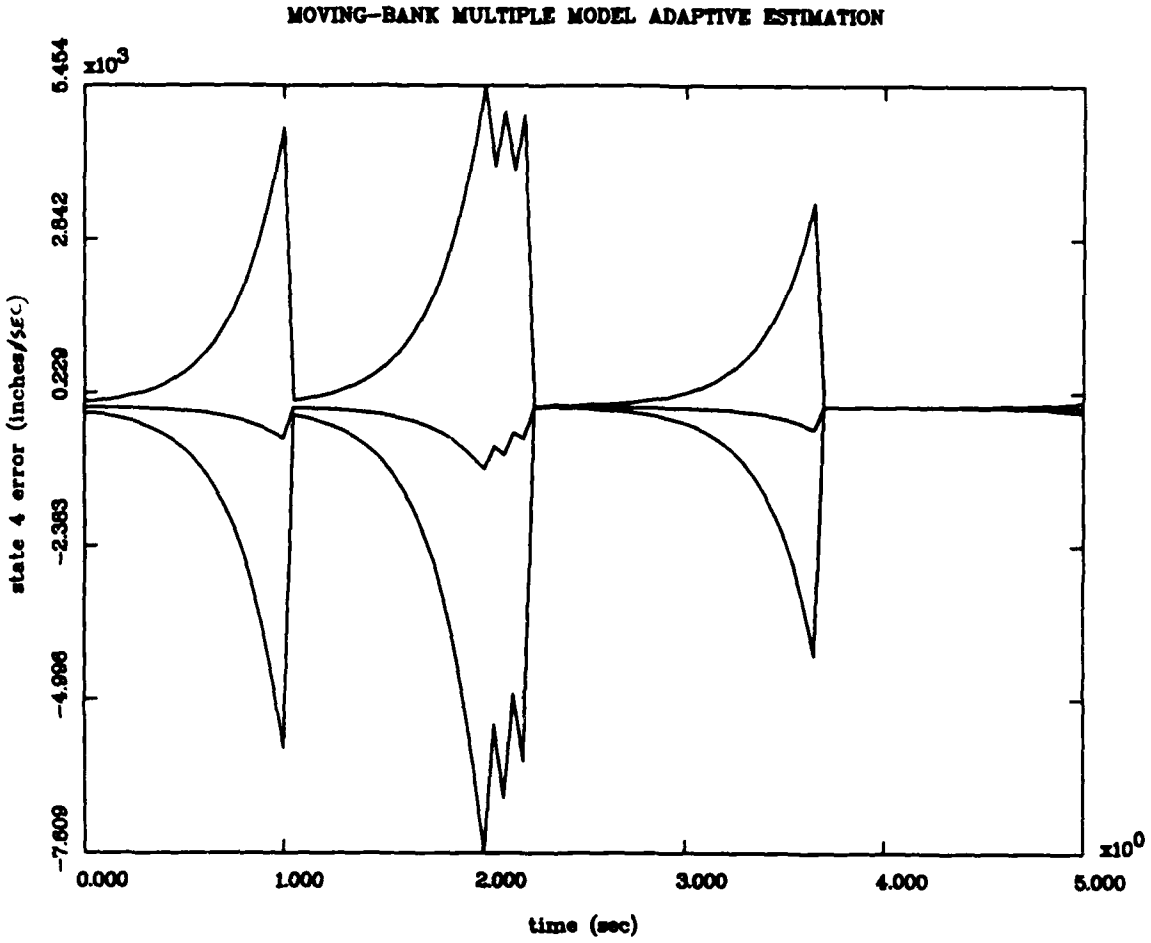

Figure 14(d) State 4
State Estimation Errors
Mean ± 1σ
Noise Level Determination Study
For R1Q1e Case
with 100 Monte Carlo Runs

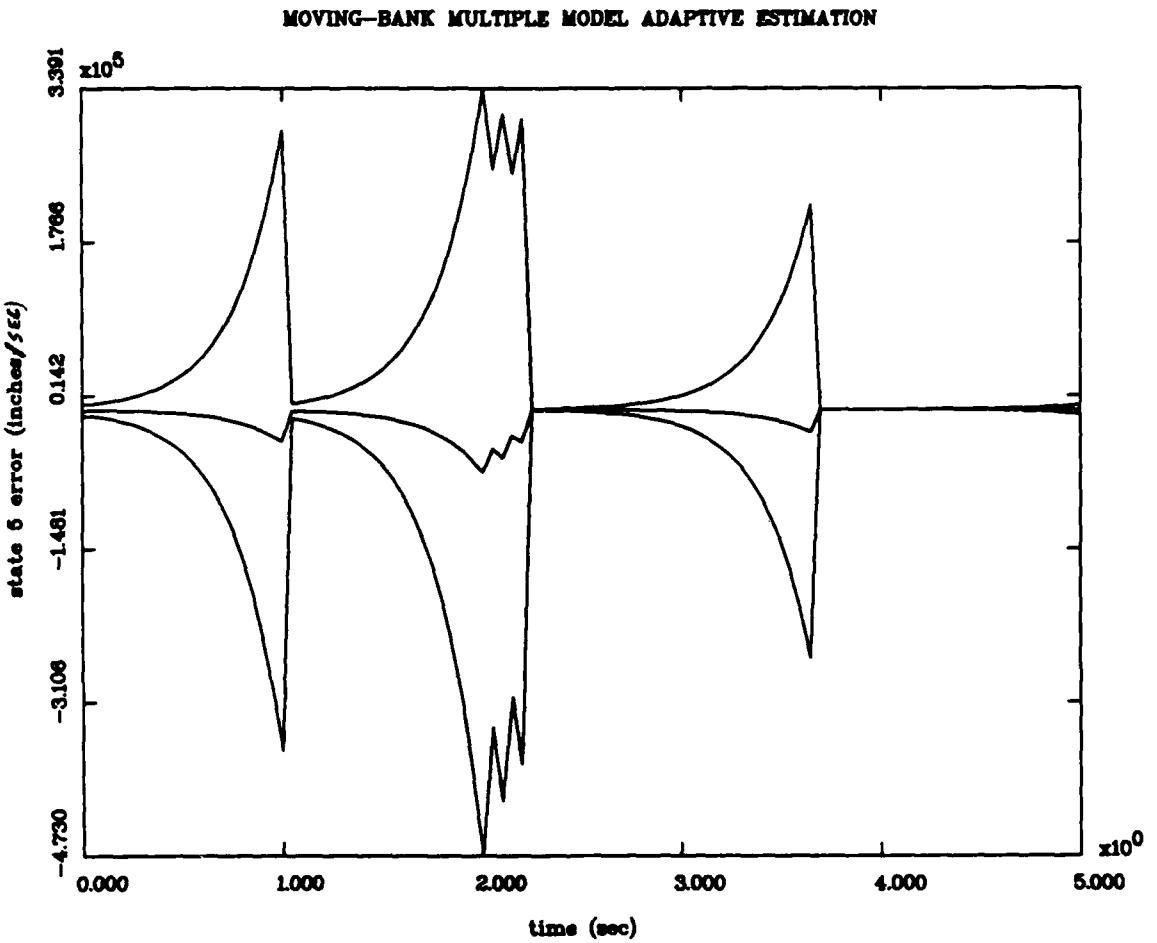

Figure 14(e) State 5
State Estimation Errors
Mean ± 1σ
Noise Level Determination Study
For R1Q1e Case
with 100 Monte Carlo Runs

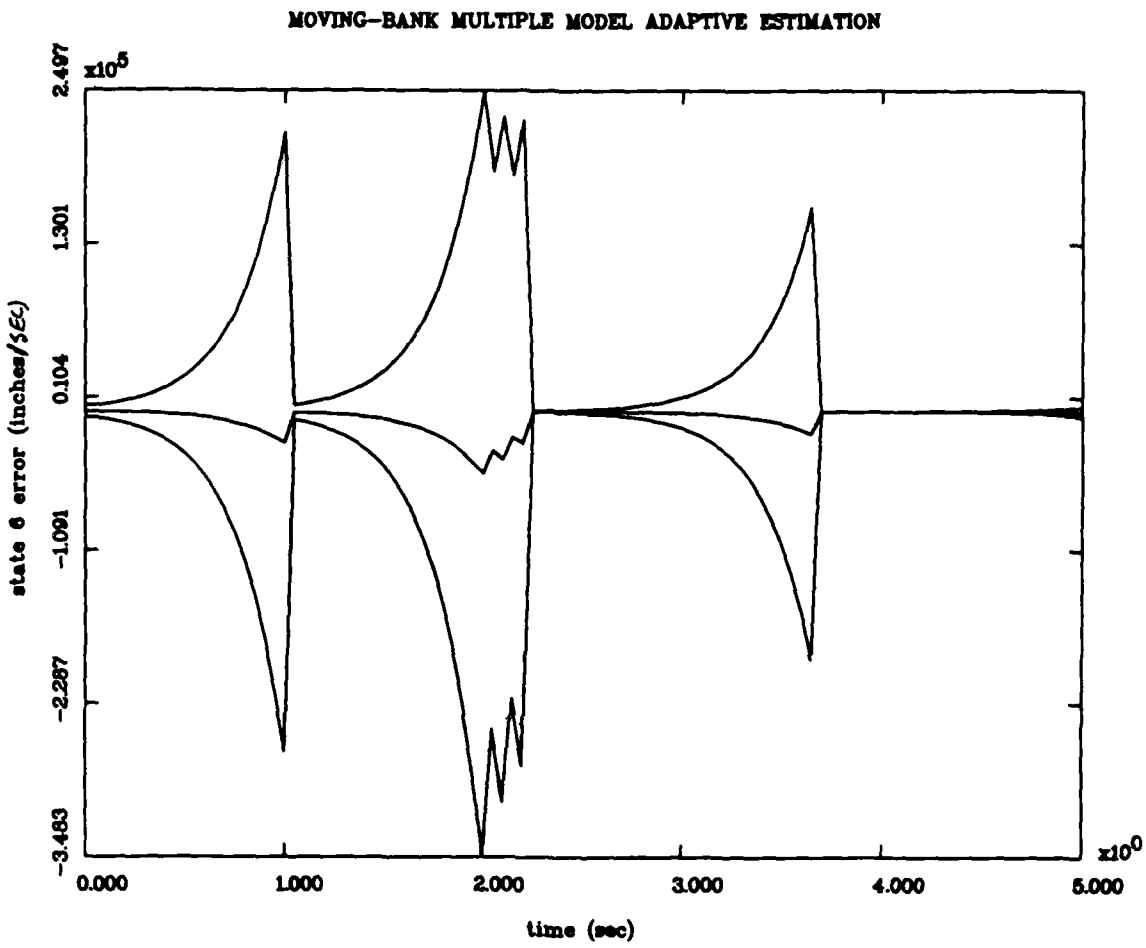

Figure 14(f) State 6
State Estimation Errors
Mean ± 1σ
Noise Level Determination Study
For R1Q1e Case
with 100 Monte Carlo Runs

During the course of the noise level determination study, the values of R_f and Q are changed by multiplying each matrix by some scalar. The object is to find some combination of the R_f and Q values which provide stable and accurate estimates of the states. The Q and R_f values and corresponding results are listed in Table 4. Throughout this study, the 6-state filters were never able to produce accurate parameter estimates. However, the filters did produce accurate and stable state estimates--which is the goal of the noise determination study.

Table 4
Q and R_f Values Used and Their Results
Maximum Position Error Standard Deviation for State 1
During the Noise Level Determination Study.
Q_{old} is as set by Lashlee [8:94],
and $R_{f\text{-}old}$ is as Determined in Section 4.4.2.

R_f Value	Q Value	Result
1 * $R_{f\text{-}old}$	2 * Q_{old}	0.5567
1 * $R_{f\text{-}old}$	5 * Q_{old}	2.6330
1 * $R_{f\text{-}old}$	10 * Q_{old}	0.5764
2 * $R_{f\text{-}old}$	5 * Q_{old}	0.6965
2 * $R_{f\text{-}old}$	10 * Q_{old}	0.4993
3 * $R_{f\text{-}old}$	10 * Q_{old}	unstable

For simplicity and convenience, the 6 cases in Table 4 will be referred by a short-hand notation which uses numbers and letters to denote the case. For example, R1Q2e stands for the first case listed in Table 4 where 1*R_f, 2*Q_{old}, and estimation errors are the distinguishing features of this case. Likewise, R1Q5e, R1Q10e, R2Q5e, R2Q10e, and R3Q10e are names for the other five cases listed in Table 4 as they are read from top to bottom,

respectively. This notation is used throughout the rest of the thesis.

Figures 15, 16, 17, 18, 19, and 20 are plots of the mean plus or minus the standard deviation of the state estimation errors from the Monte Carlo simulations given in Table 4. As stated previously, the basic idea of this noise level determination study is to "inform" the filter model that it does not have an adequate representation of the real world (truth model), by increasing its measurement noise covariance matrix (R_f) to account for the unmodeled higher order bending mode effects. However, as the results indicate, increasing the R_f alone is not enough. When R_f alone is increased, the Kalman filter gain, given by Equation (21), decreases. This decrease in the Kalman filter gain is undesirable. Therefore, the Q matrix ($Q_f = Q_t$) must also be increased to keep the Kalman filter gain from decreasing. The relationship between the Kalman filter gain and the Q_f and R_f values is easier to understand for the scalar case where the gain is the ratio Q_f/R_f.

The first plot of Figure 15 (R1Q2e), like that of Figure 13, appears unstable while the other plots of Figure 15 appear stable. This also merits further investigation by increasing the number of Monte Carlo runs to 100. The plots of Figure 21 are the results. The standard deviation peaks to a single very large value. Again, this peak is probably due to the inability of the

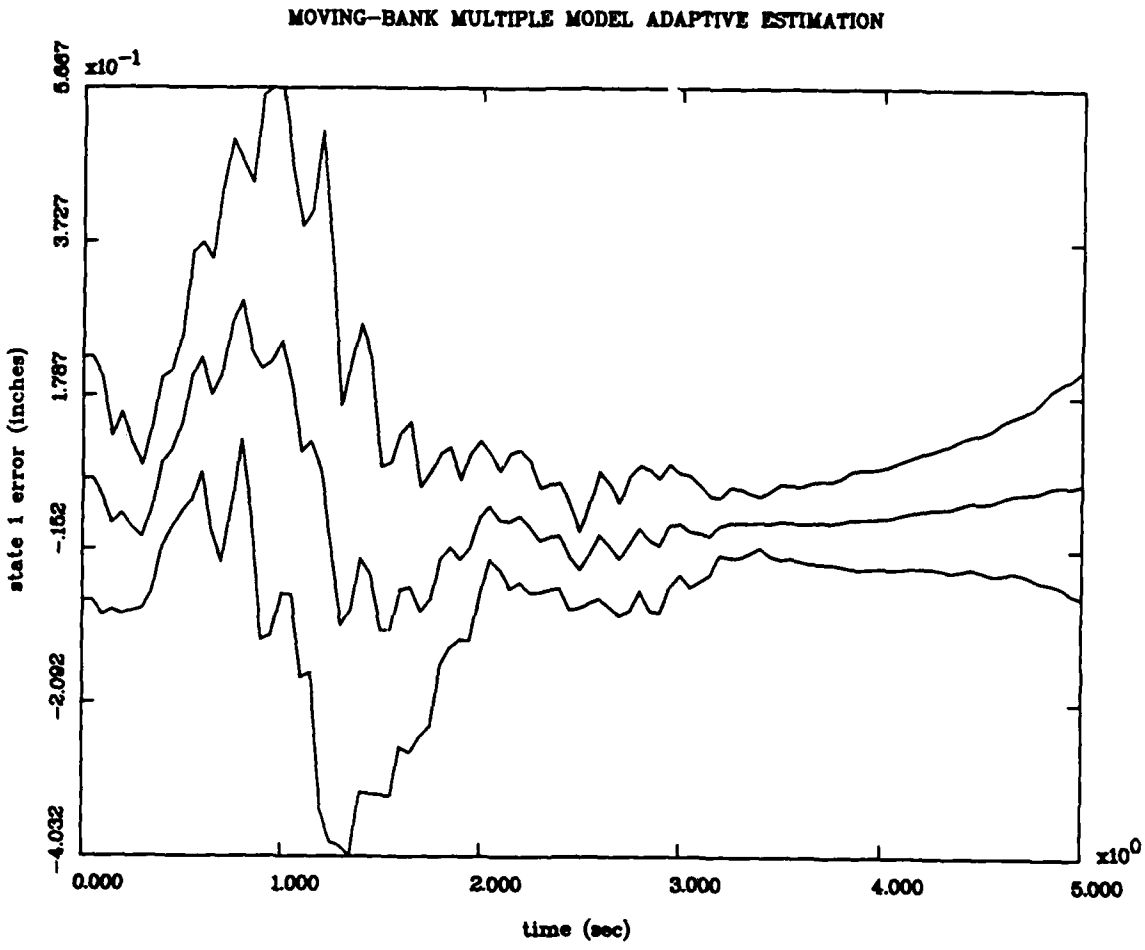

Figure 15(a) State 1
State Estimation Errors
Mean ± 1σ
Noise Level Determination Study
For R1Q2e Case
with 10 Monte Carlo Runs

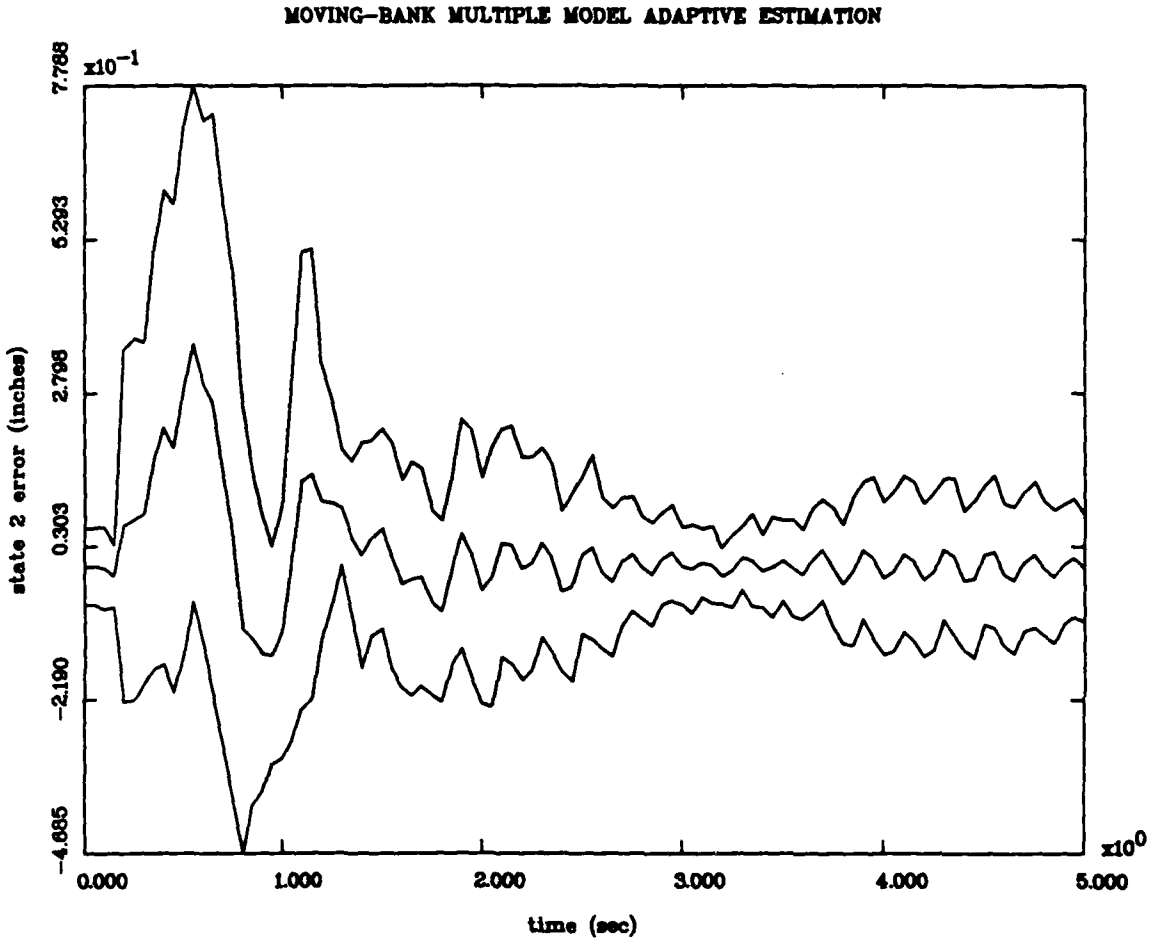

Figure 15(b) State 2
State Estimation Errors
Mean ± 1σ
Noise Level Determination Study
For R1Q2e Case
with 10 Monte Carlo Runs

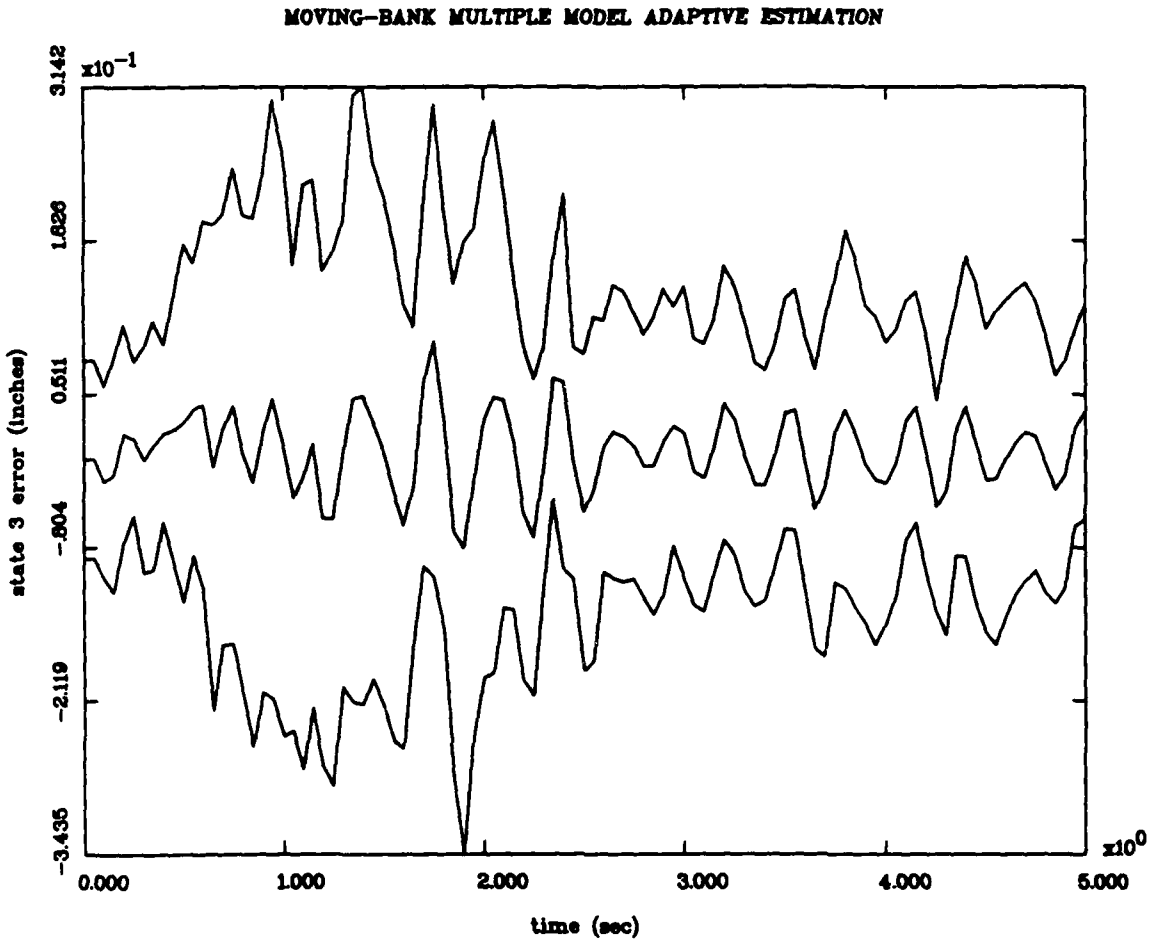

Figure 15(c) State 3
State Estimation Errors
Mean ± 1σ
Noise Level Determination Study
For R1Q2e Case
with 10 Monte Carlo Runs

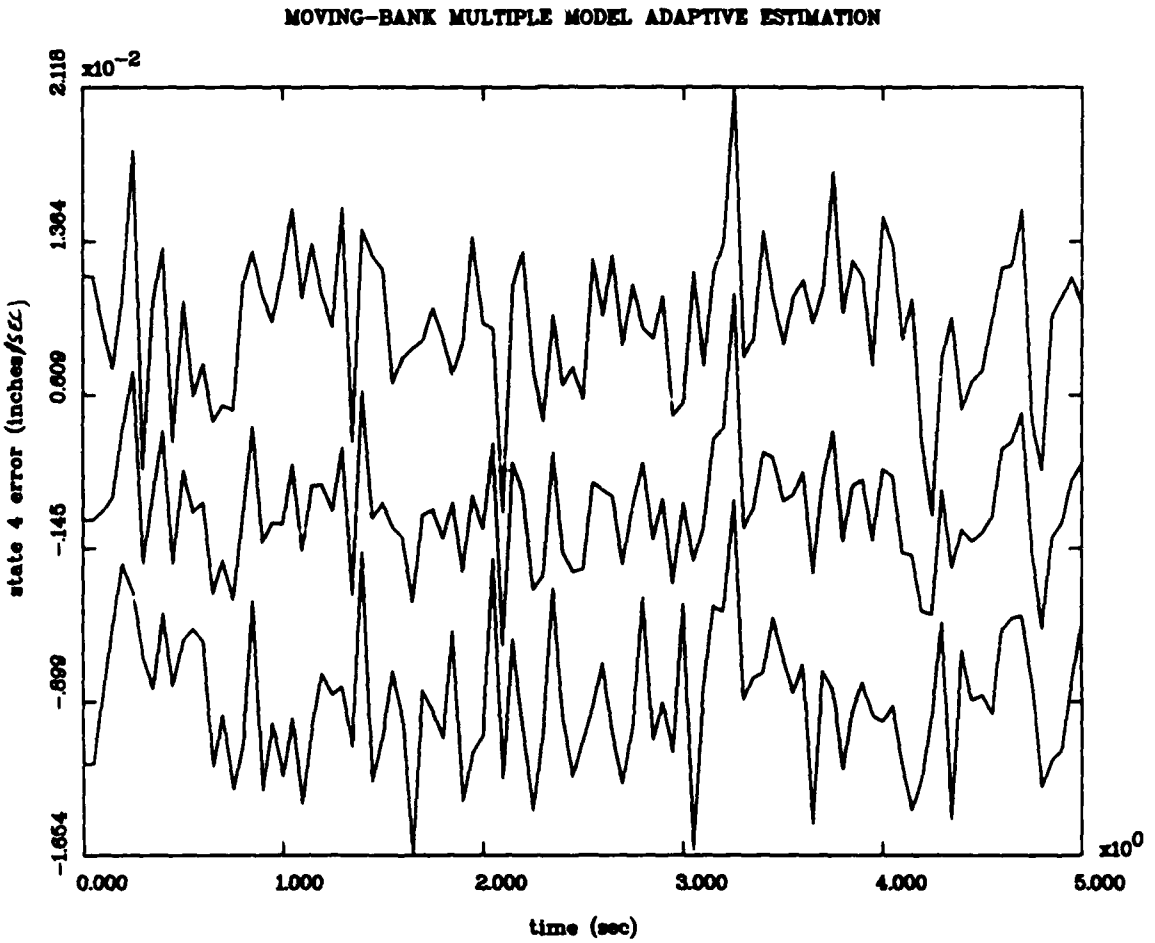

Figure 15(d) State 4
State Estimation Errors
Mean ± 1σ
Noise Level Determination Study
For R1Q2e Case
with 10 Monte Carlo Runs

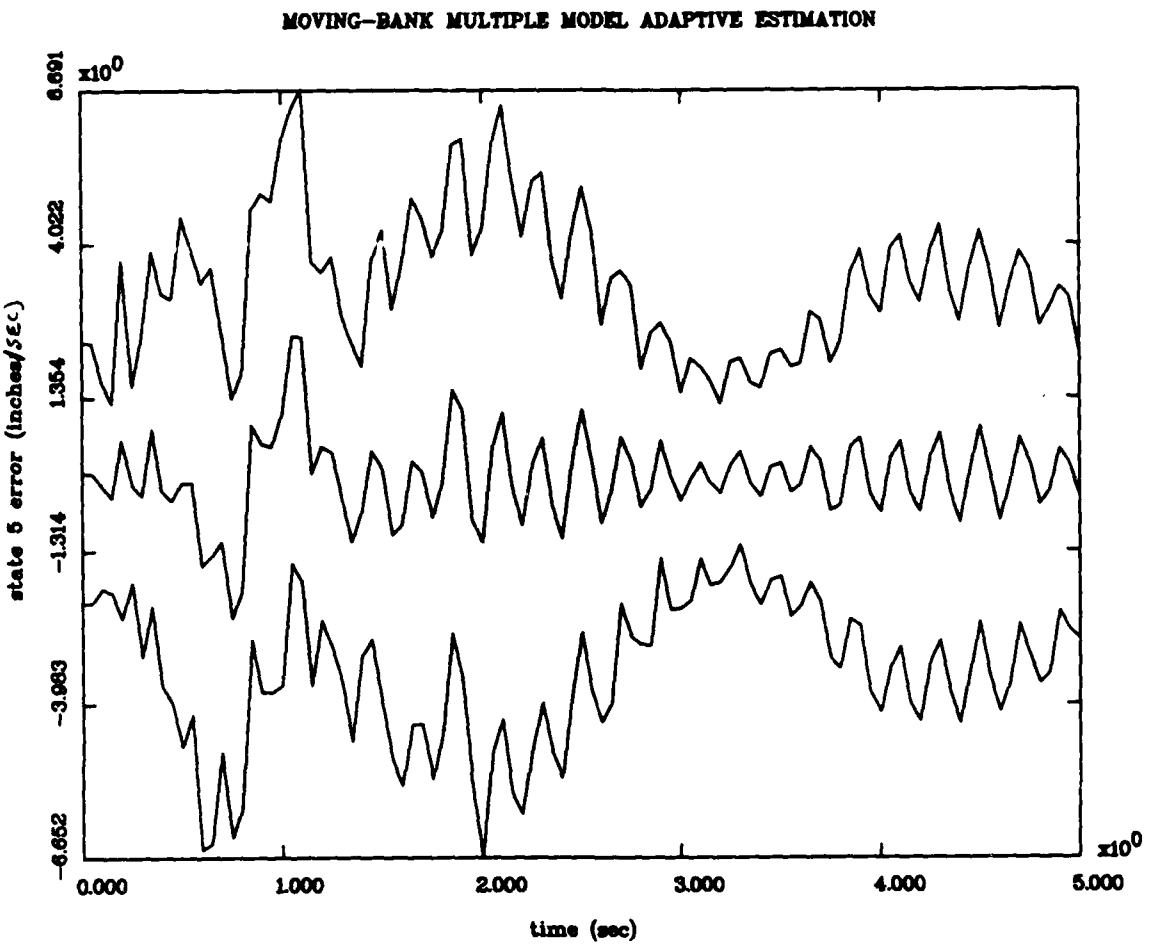

Figure 15(e) State 5
State Estimation Errors
Mean ± 1σ
Noise Level Determination Study
For R1Q2e Case
with 10 Monte Carlo Runs

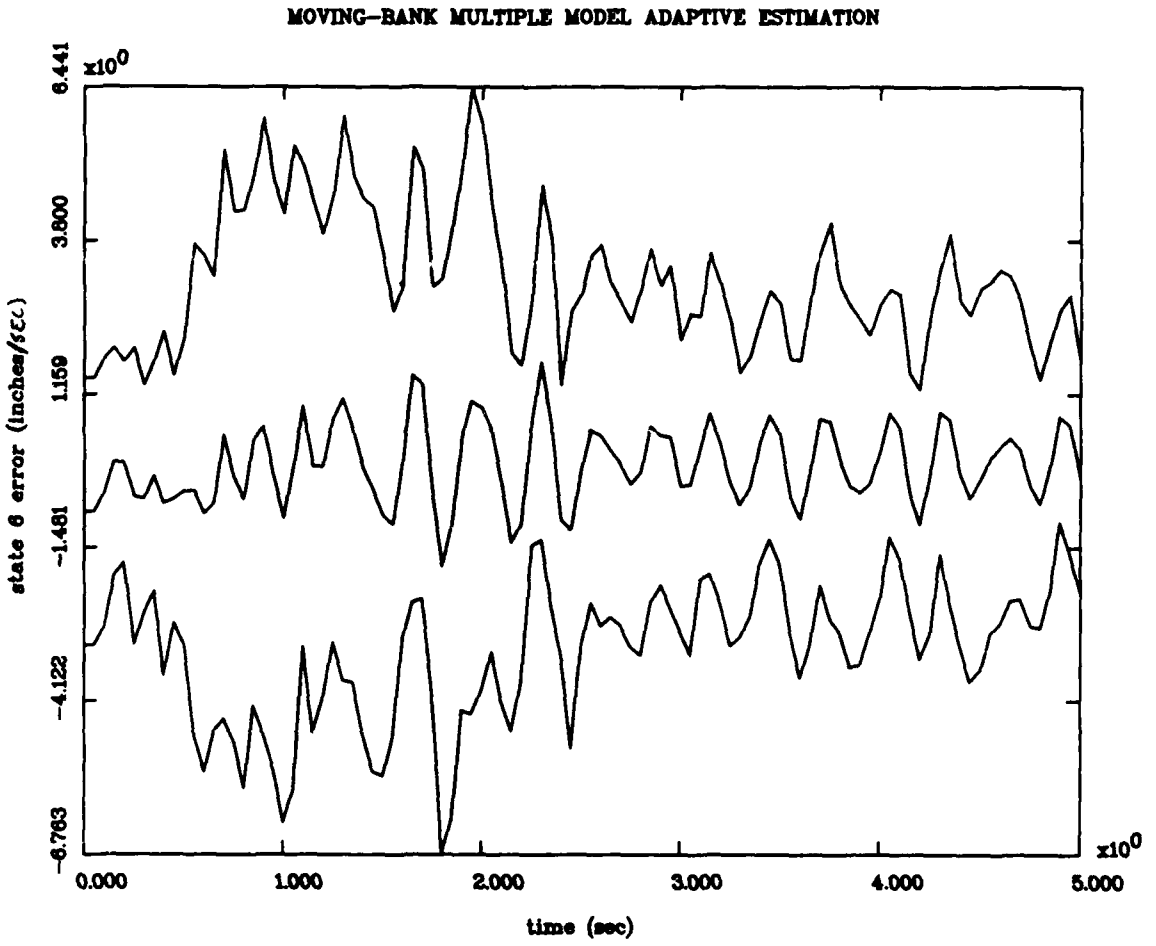

Figure 15(f) State 6
State Estimation Errors
Mean ± 1σ
Noise Level Determination Study
For R1Q2e Case
with 10 Monte Carlo Runs

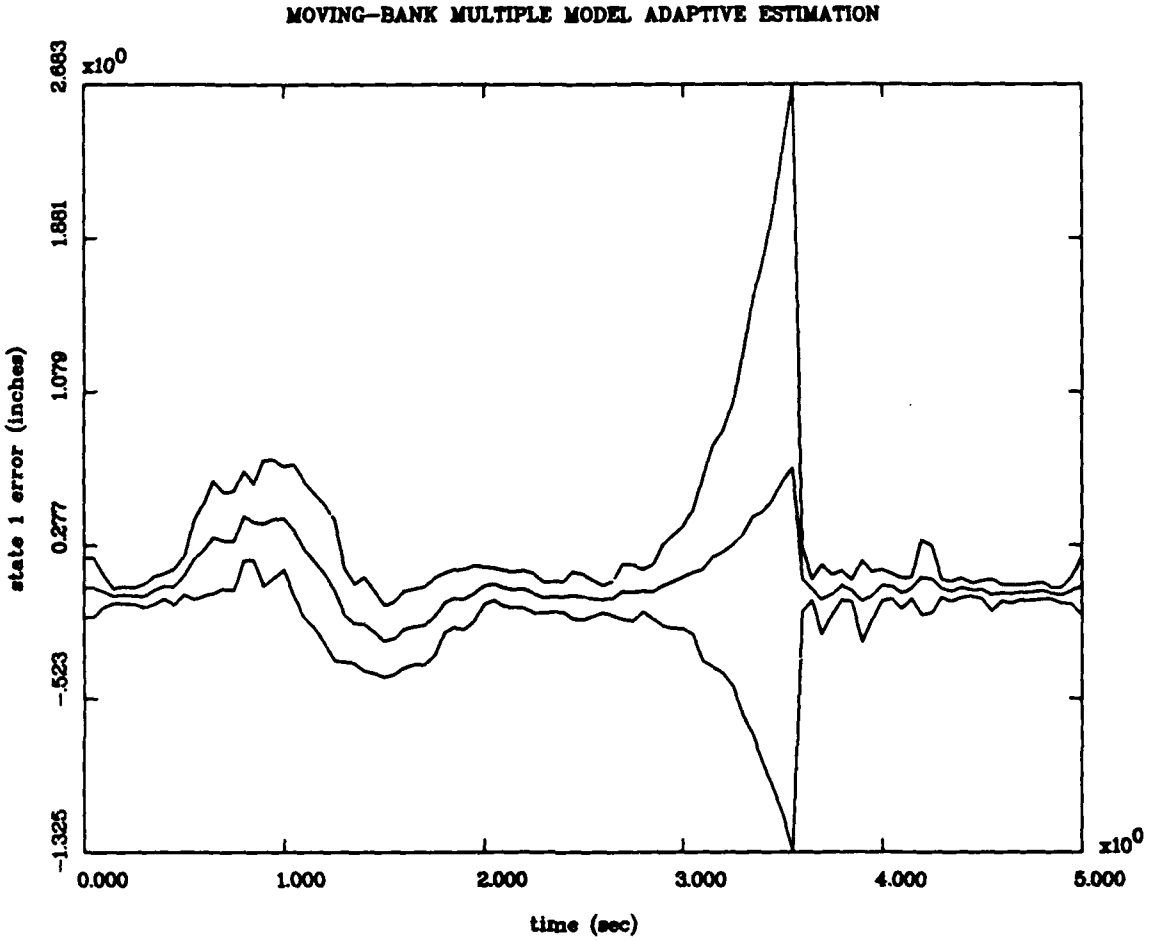

Figure 16(a) State 1
State Estimation Errors
Mean ± 1σ
Noise Level Determination Study
For R1Q5e Case
with 10 Monte Carlo Runs

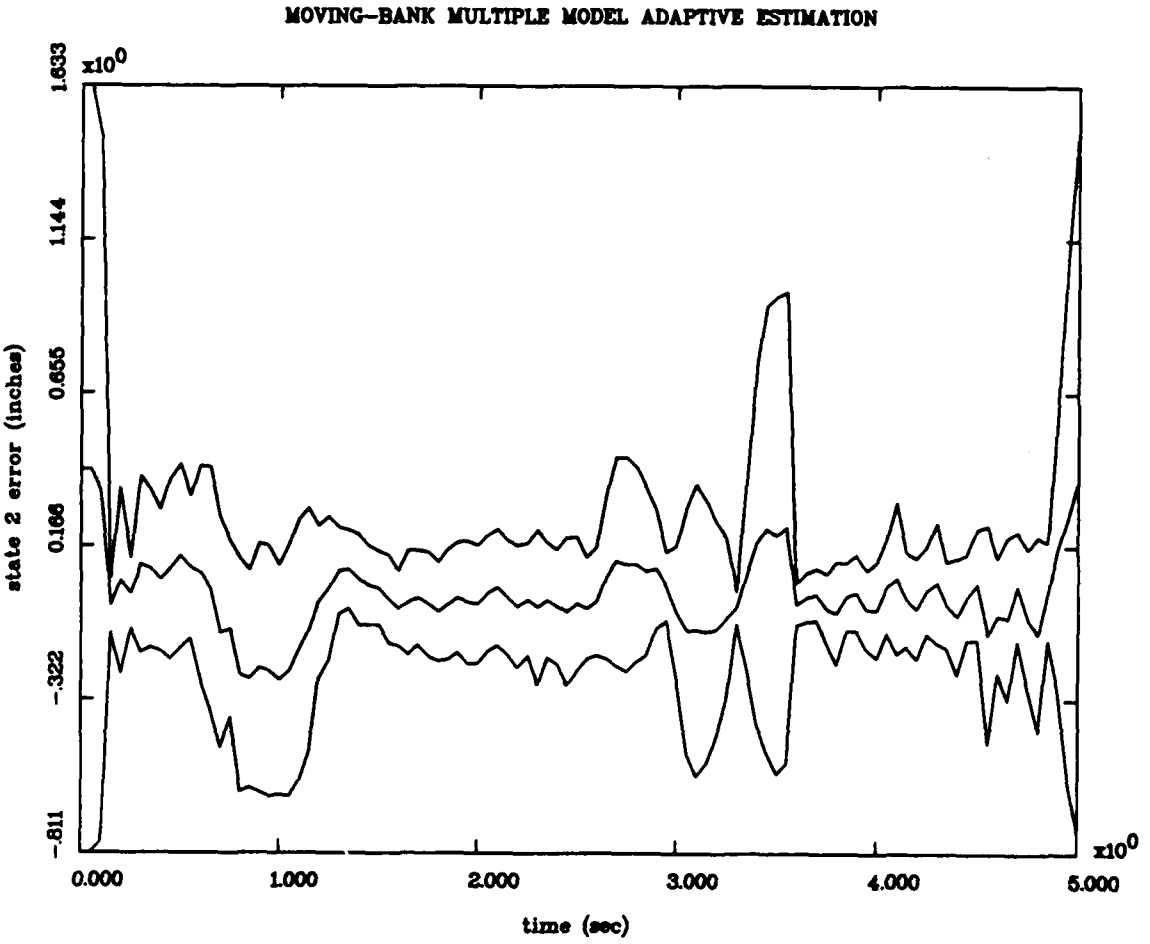

Figure 16(b) State 2
State Estimation Errors
Mean ± 1σ
Noise Level Determination Study
For R1Q5e Case
with 10 Monte Carlo Runs

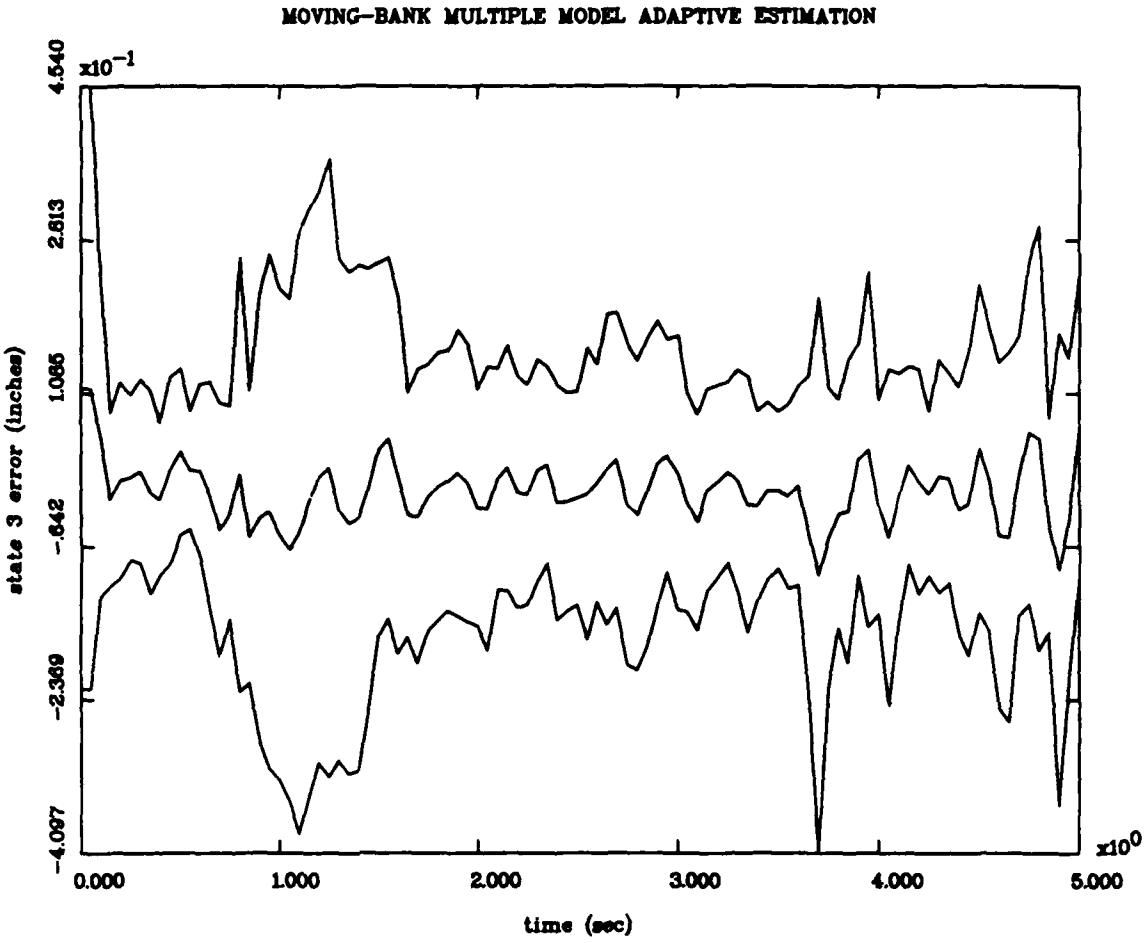

Figure 16(c) State 3
State Estimation Errors
Mean ± 1σ
Noise Level Determination Study
For R1Q5e Case
with 10 Monte Carlo Runs

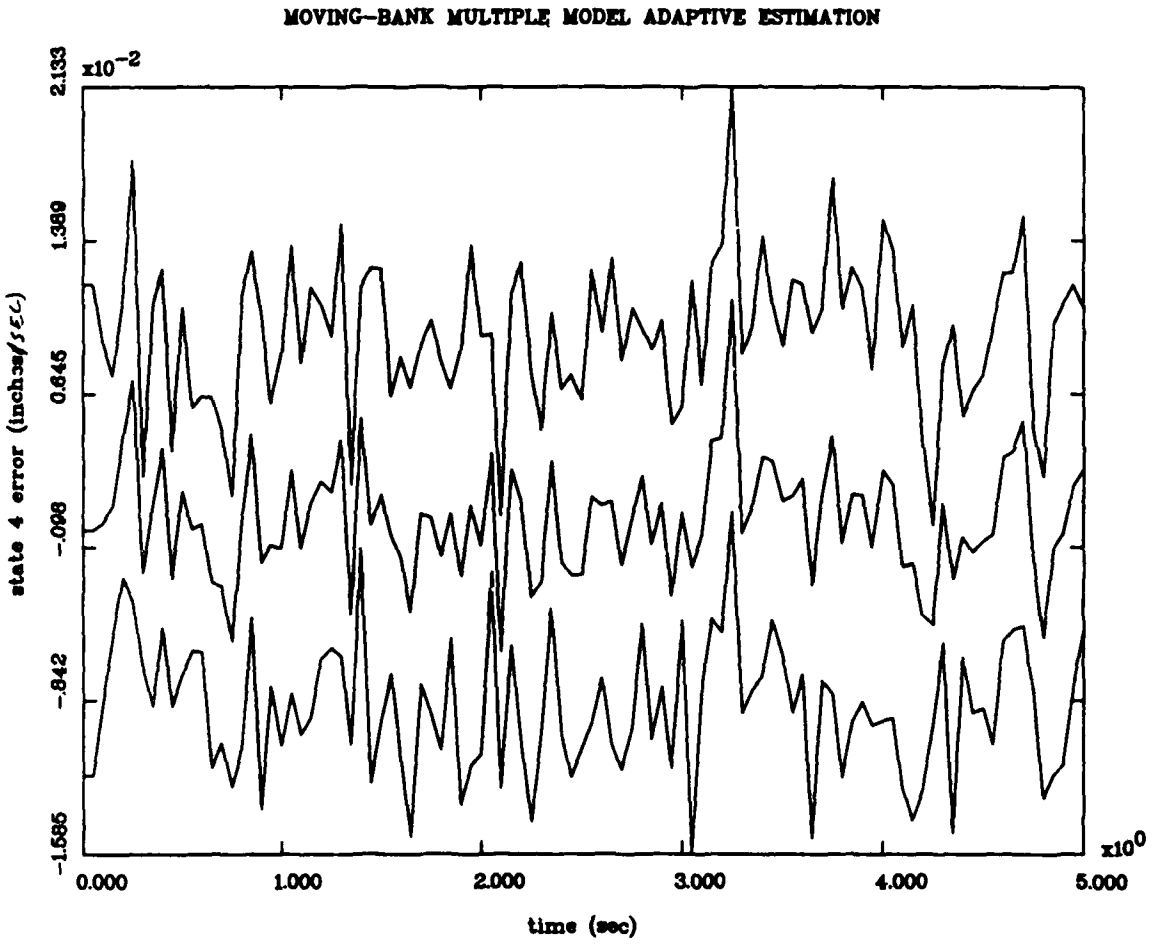

Figure 16(d) State 4
State Estimation Errors
Mean ± 1σ
Noise Level Determination Study
For R1Q5e Case
with 10 Monte Carlo Runs

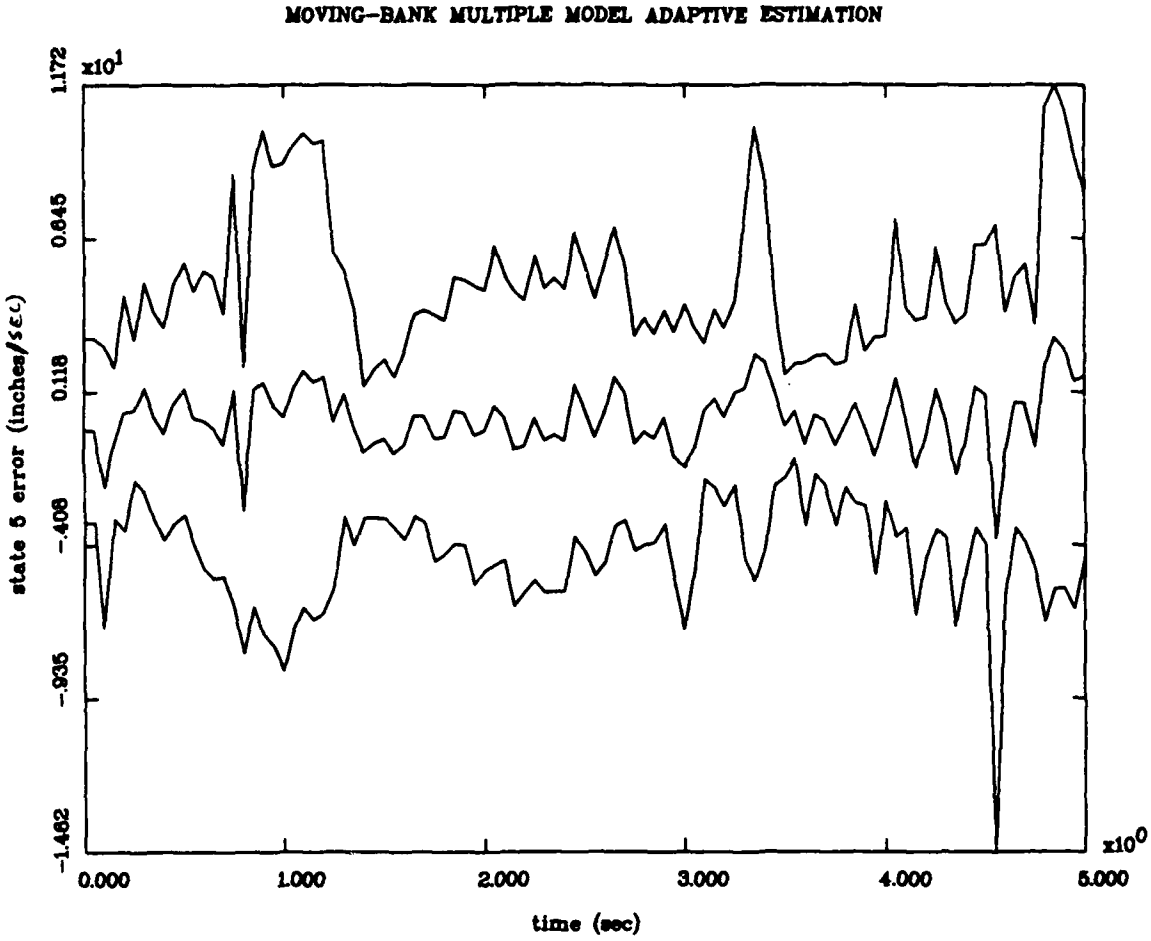

Figure 16(e) State 5
State Estimation Errors
Mean ± 1σ
Noise Level Determination Study
For R1Q5e Case
with 10 Monte Carlo Runs

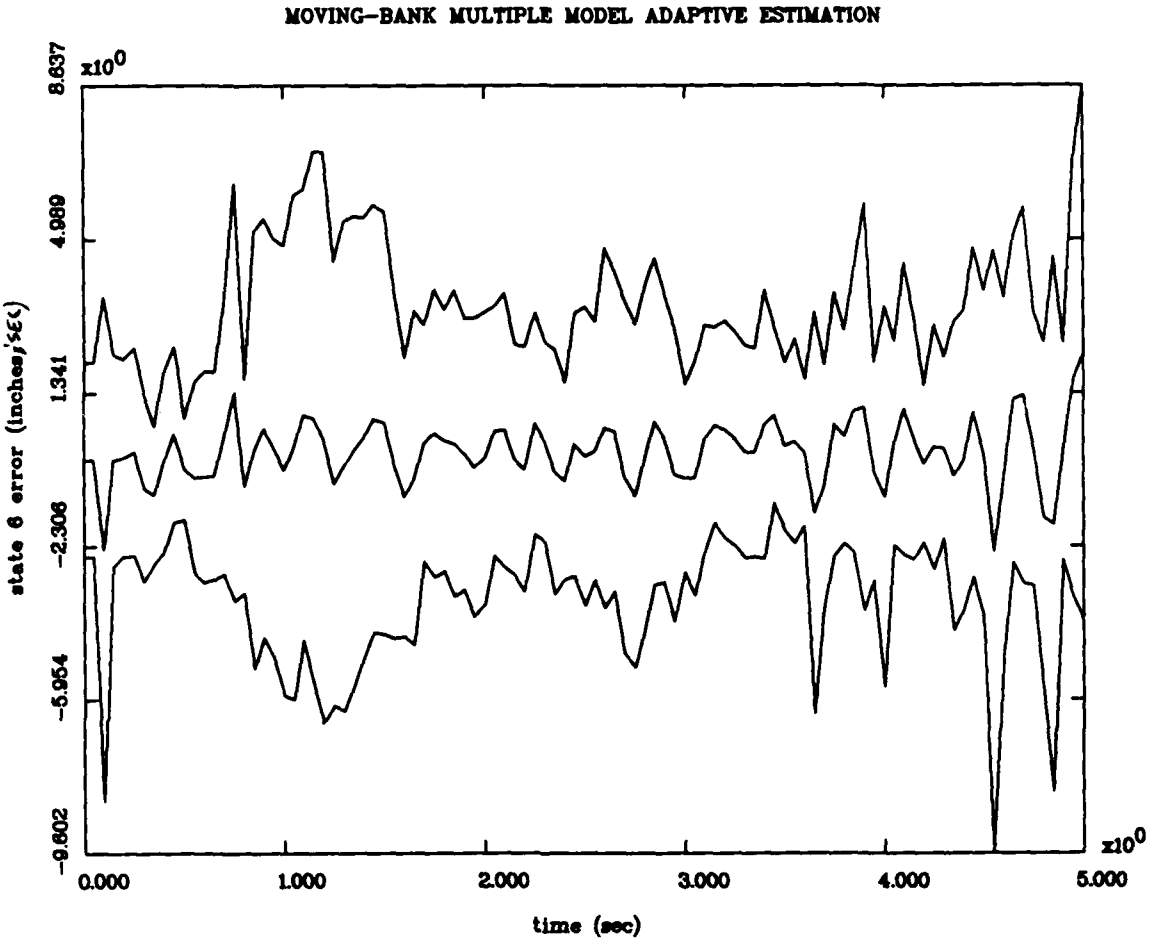

Figure 16(f) State 6
State Estimation Errors
Mean ± 1σ
Noise Level Determination Study
For R1Q5e Case
with 10 Monte Carlo Runs

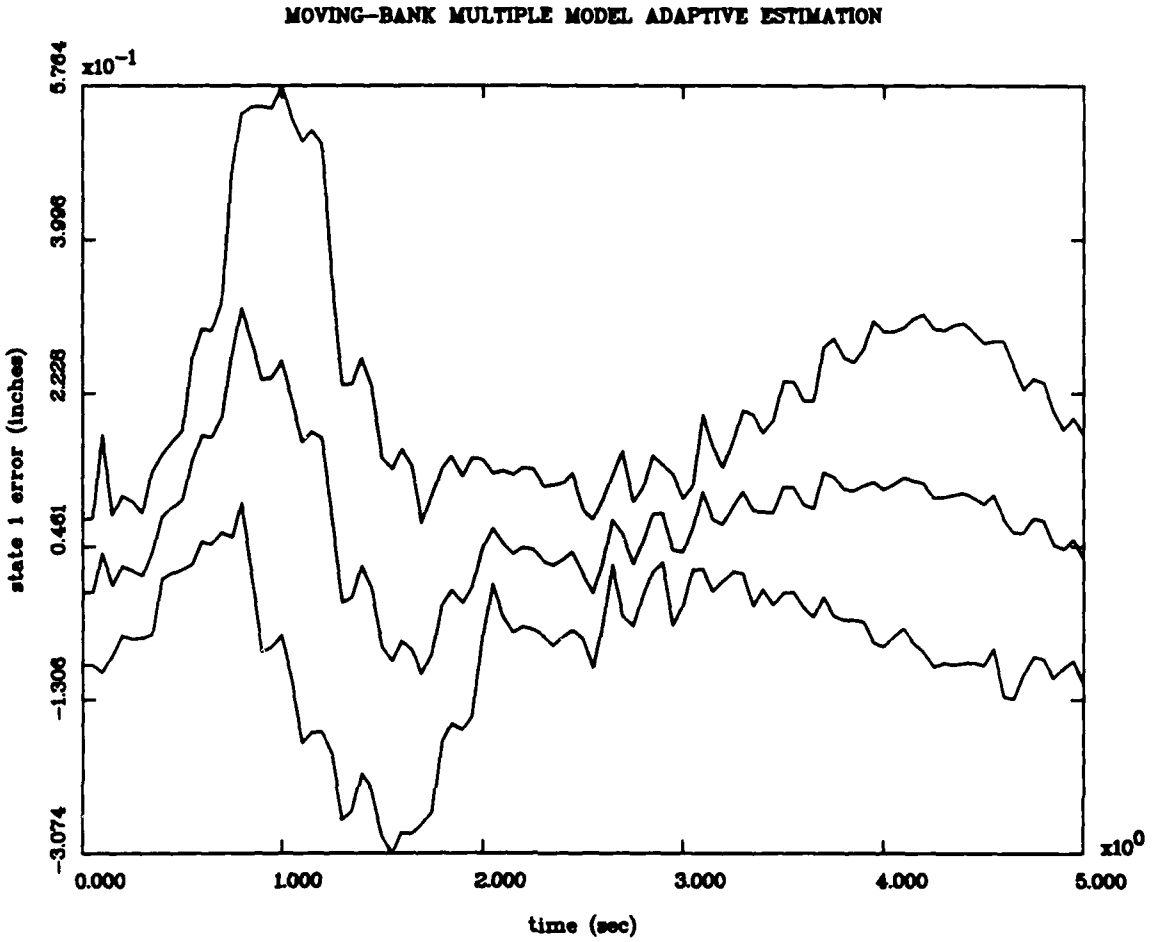

Figure 17(a) State 1
State Estimation Errors
Mean ± 1σ
Noise Level Determination Study
For R1Q10e Case
with 10 Monte Carlo Runs

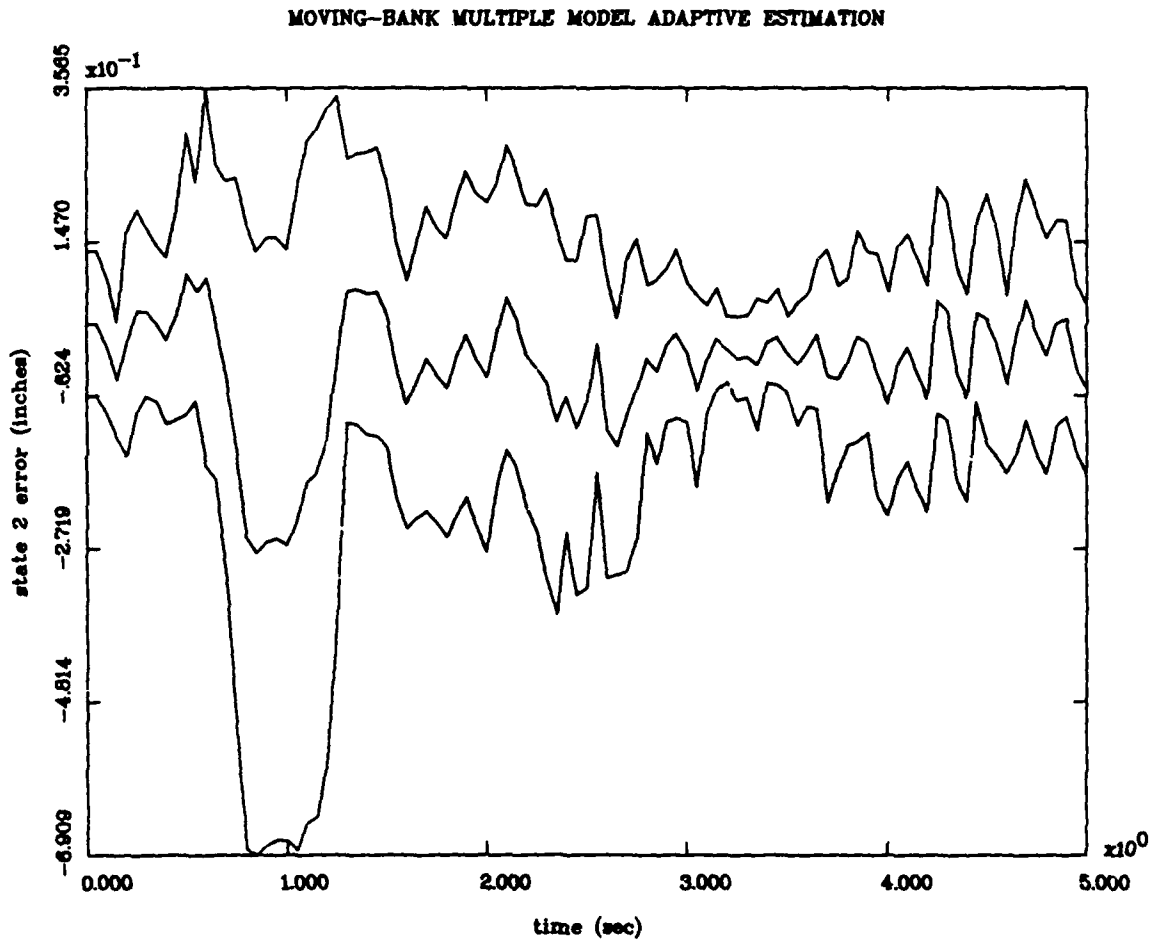

Figure 17(b) State 2
State Estimation Errors
Mean ± 1σ
Noise Level Determination Study
For R1Q10e Case
with 10 Monte Carlo Runs

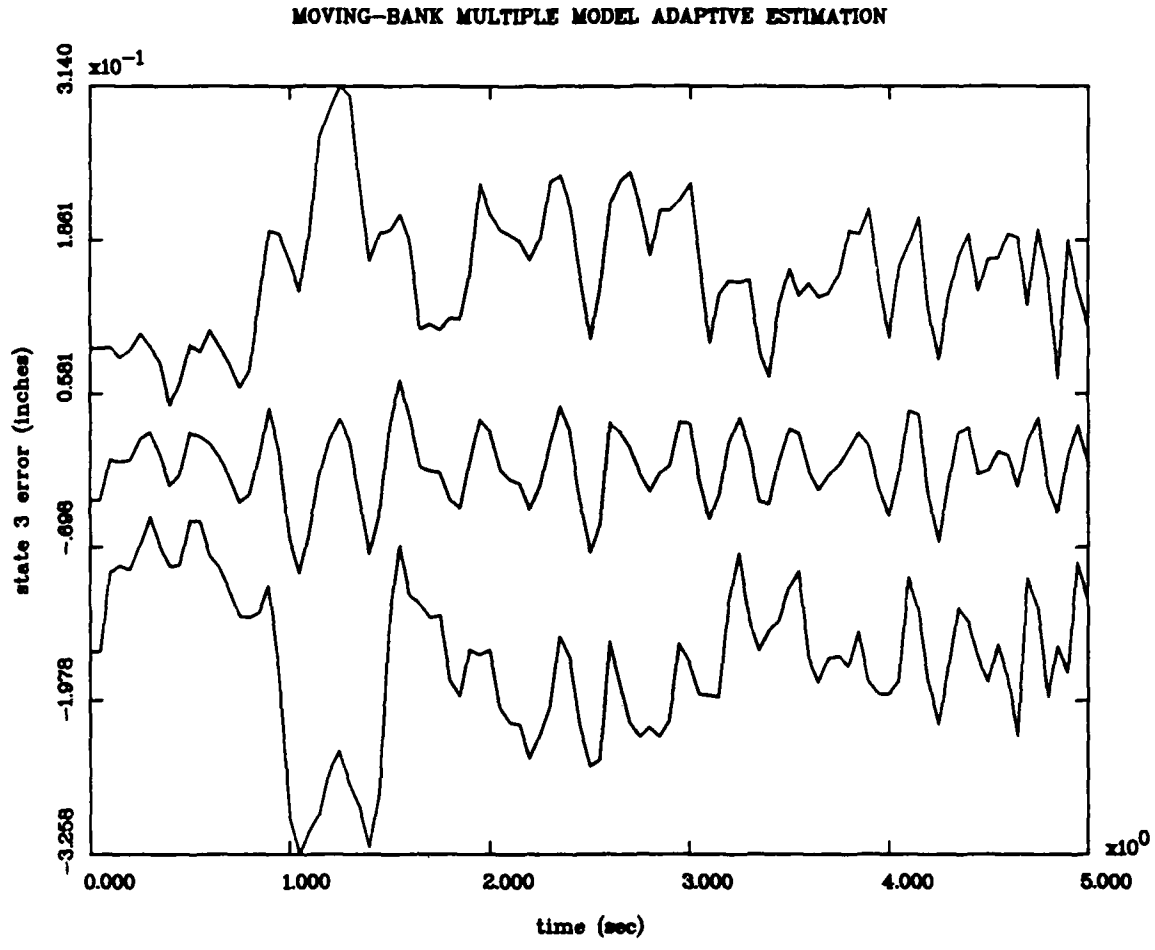

Figure 17(c) State 3
State Estimation Errors
Mean ± 1σ
Noise Level Determination Study
For R1Q10e Case
with 10 Monte Carlo Runs

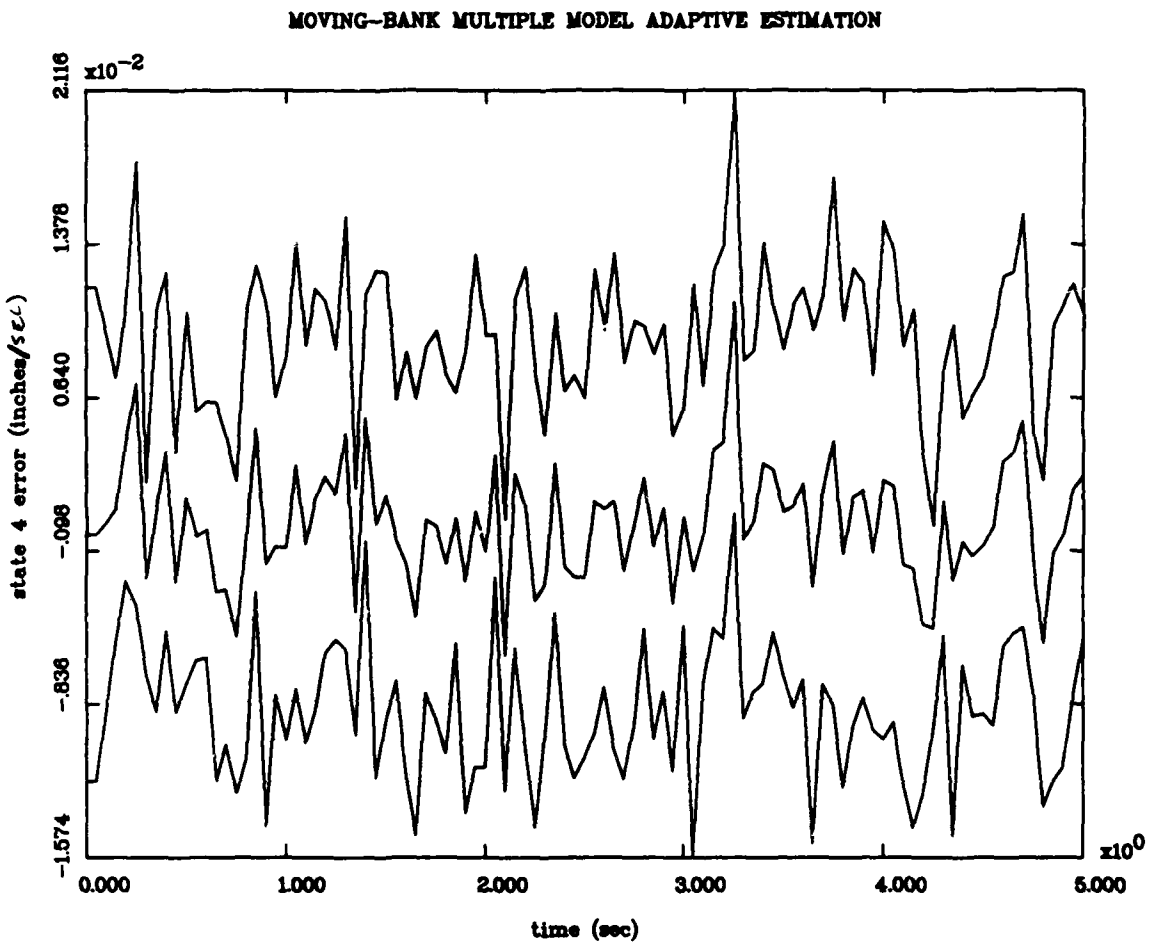

Figure 17(d) State 4
State Estimation Errors
Mean ± 1σ
Noise Level Determination Study
For R1Q10e Case
with 10 Monte Carlo Runs

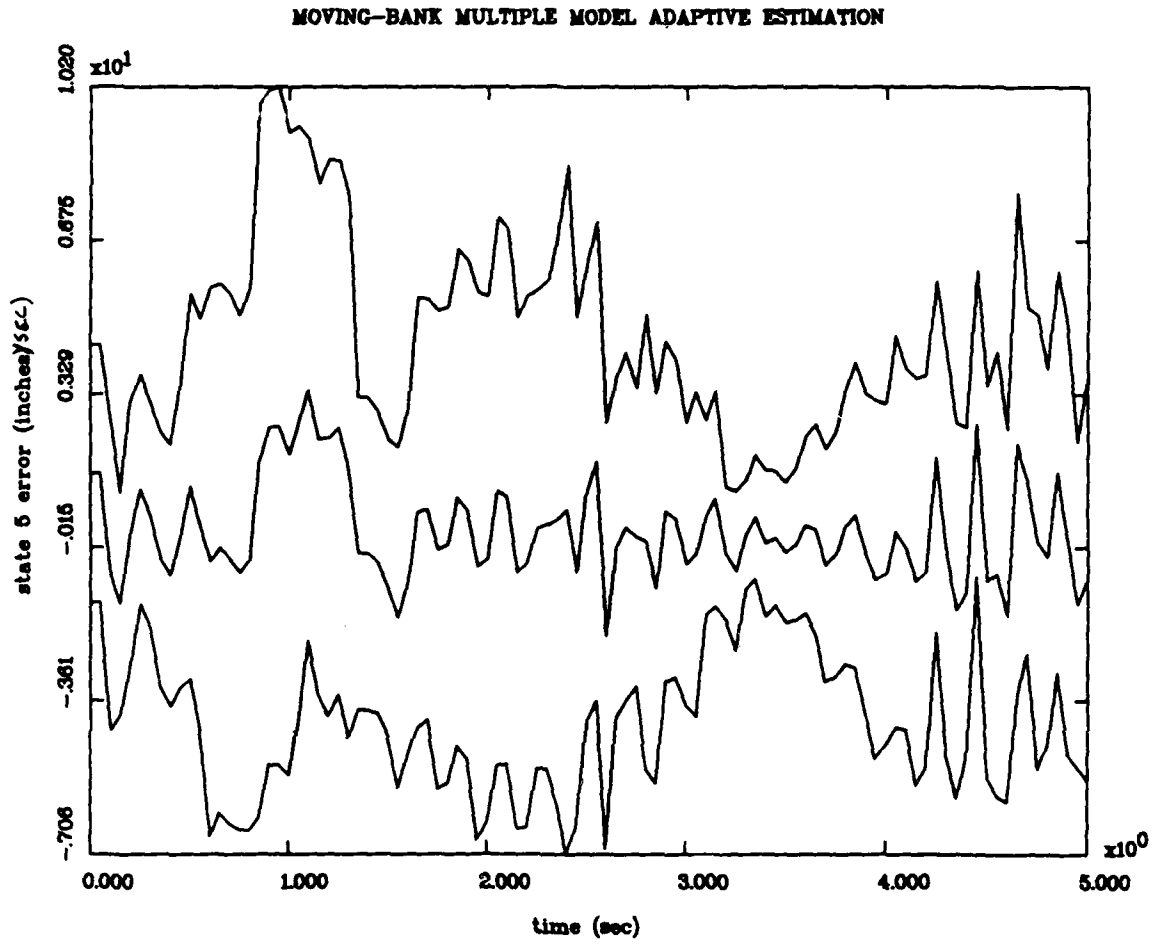

Figure 17(e) State 5
State Estimation Errors
Mean ± 1σ
Noise Level Determination Study
For R1Q10e Case
with 10 Monte Carlo Runs

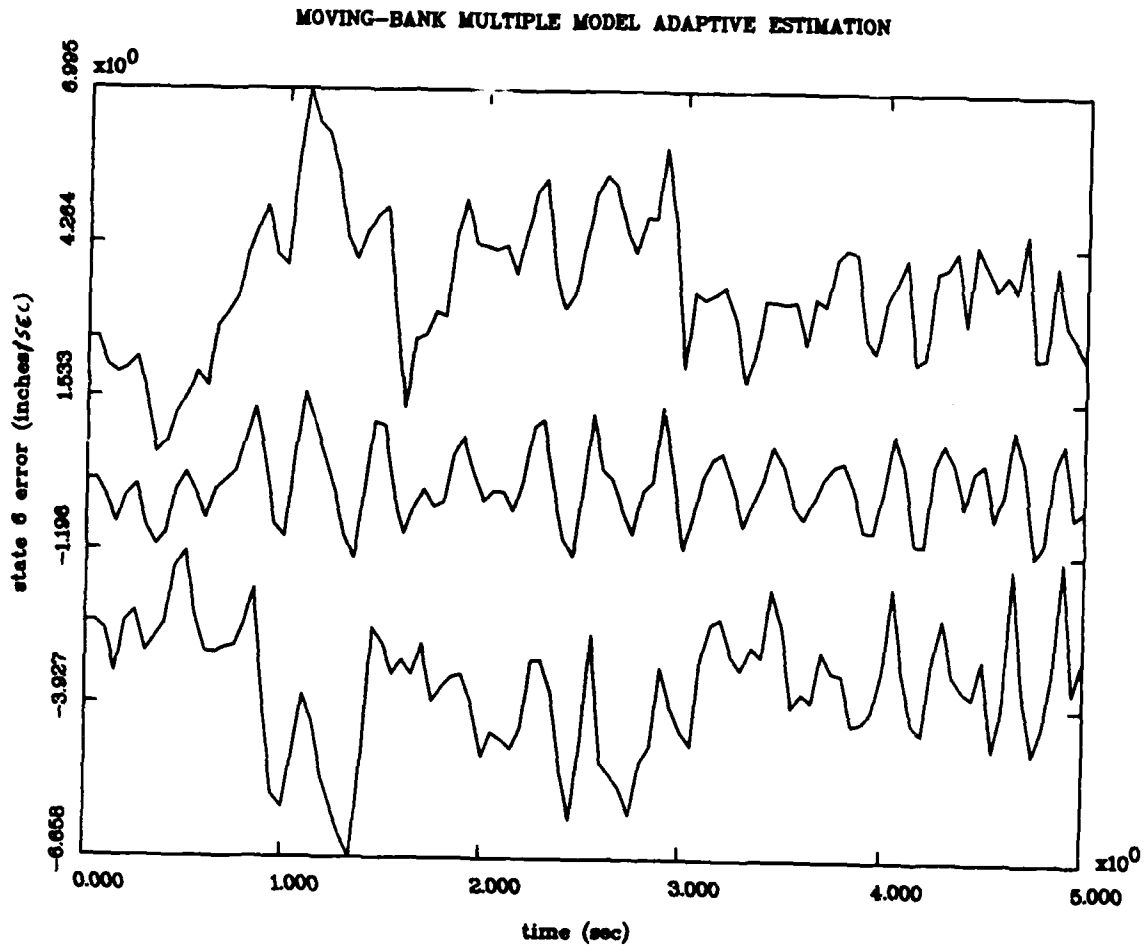

Figure 17(f) State 6
State Estimation Errors
Mean ± 1σ
Noise Level Determination Study
For R1Q10e Case
with 10 Monte Carlo Runs

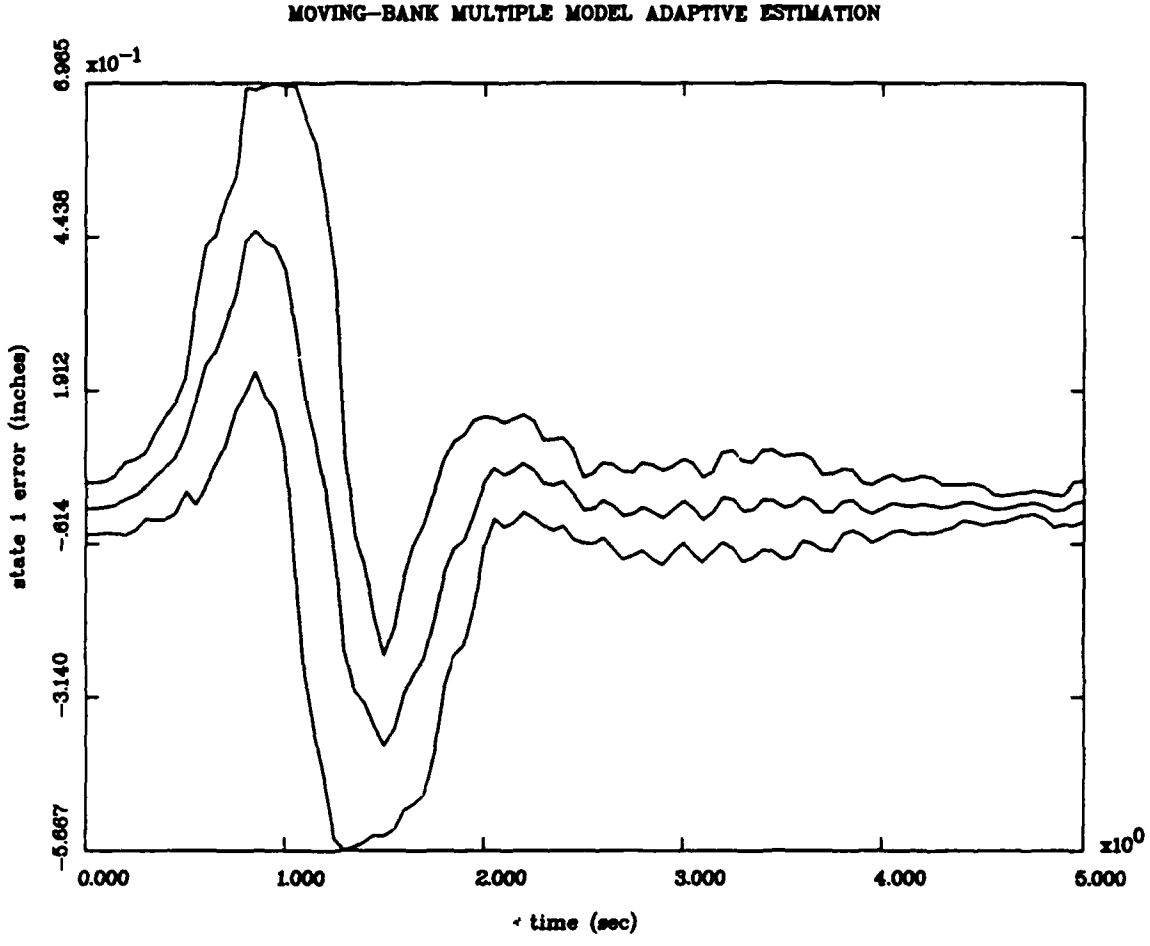

Figure 18(a) State 1
State Estimation Errors
Mean ± 1σ
Noise Level Determination Study
For R2Q5e Case
with 10 Monte Carlo Runs

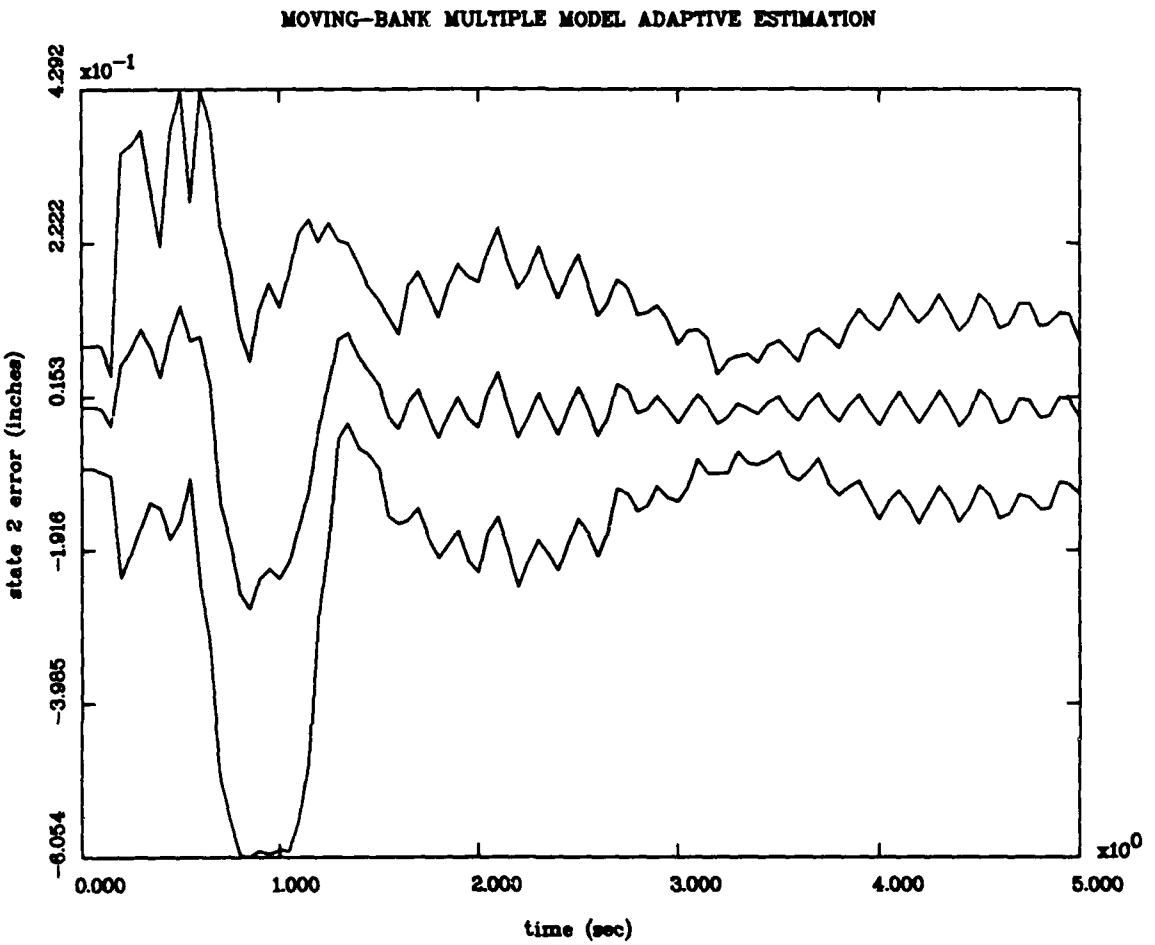

Figure 18(b) State 2
State Estimation Errors
Mean ± 1σ
Noise Level Determination Study
For R2Q5e Case
with 10 Monte Carlo Runs

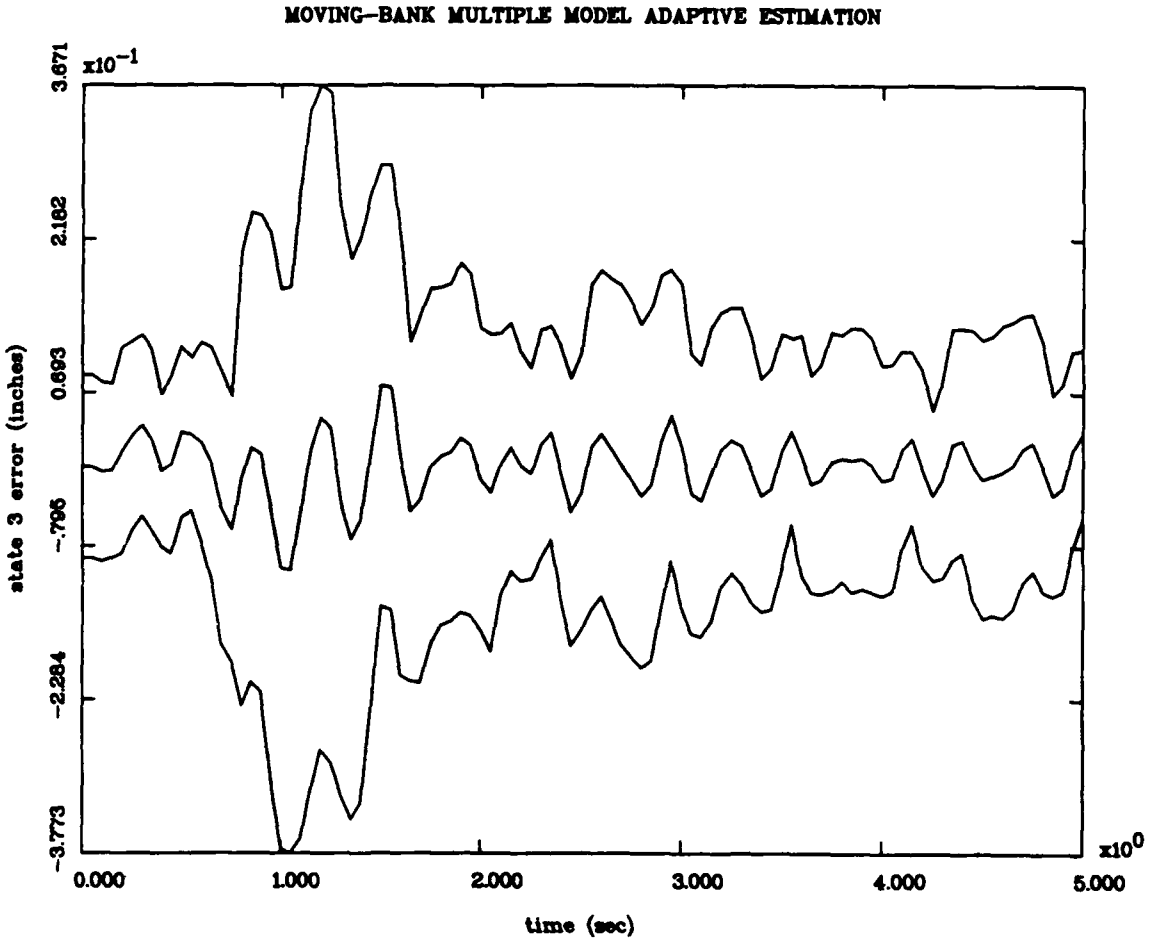

Figure 18(c) State 3
State Estimation Errors
Mean ± 1σ
Noise Level Determination Study
For R2Q5e Case
with 10 Monte Carlo Runs

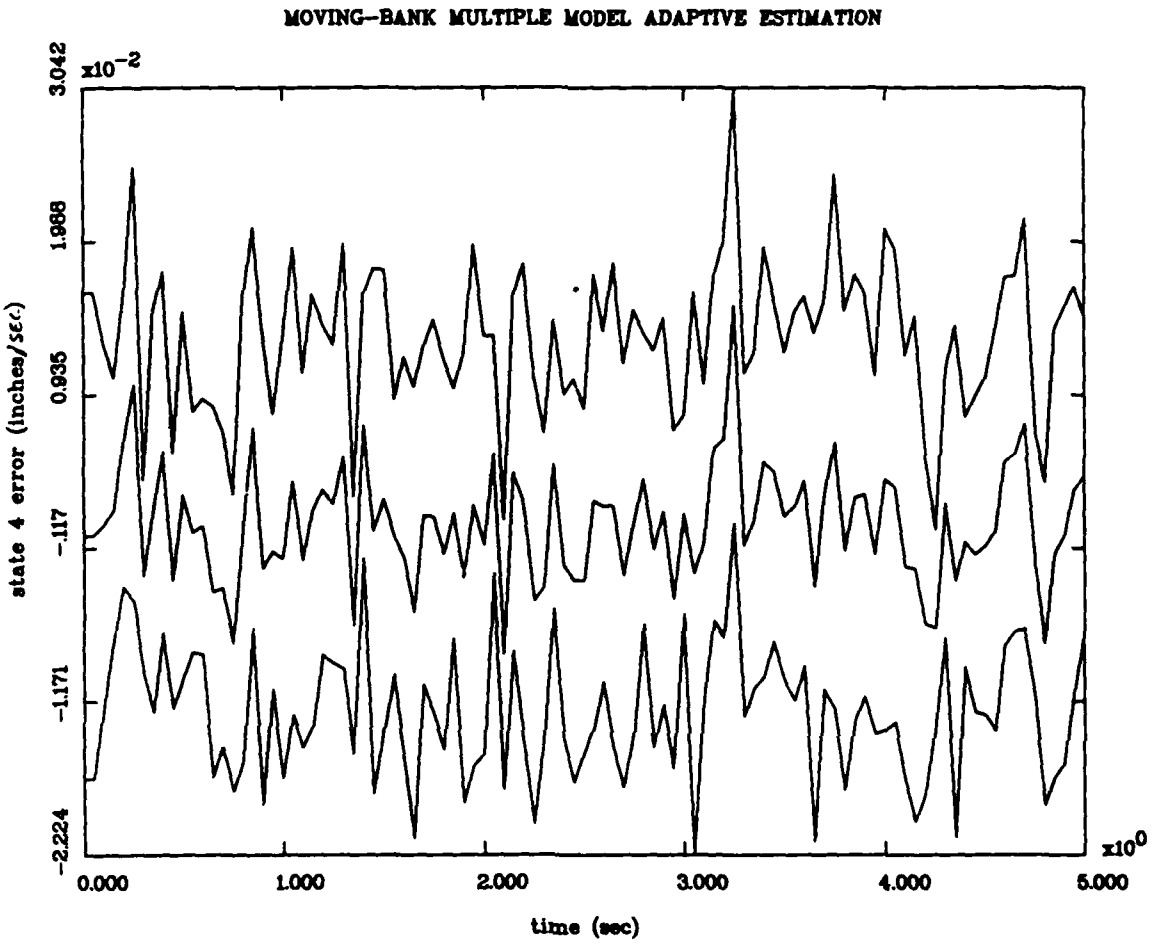

Figure 18(d) State 4
State Estimation Errors
Mean ± 1σ
Noise Level Determination Study
For R2Q5e Case
with 10 Monte Carlo Runs

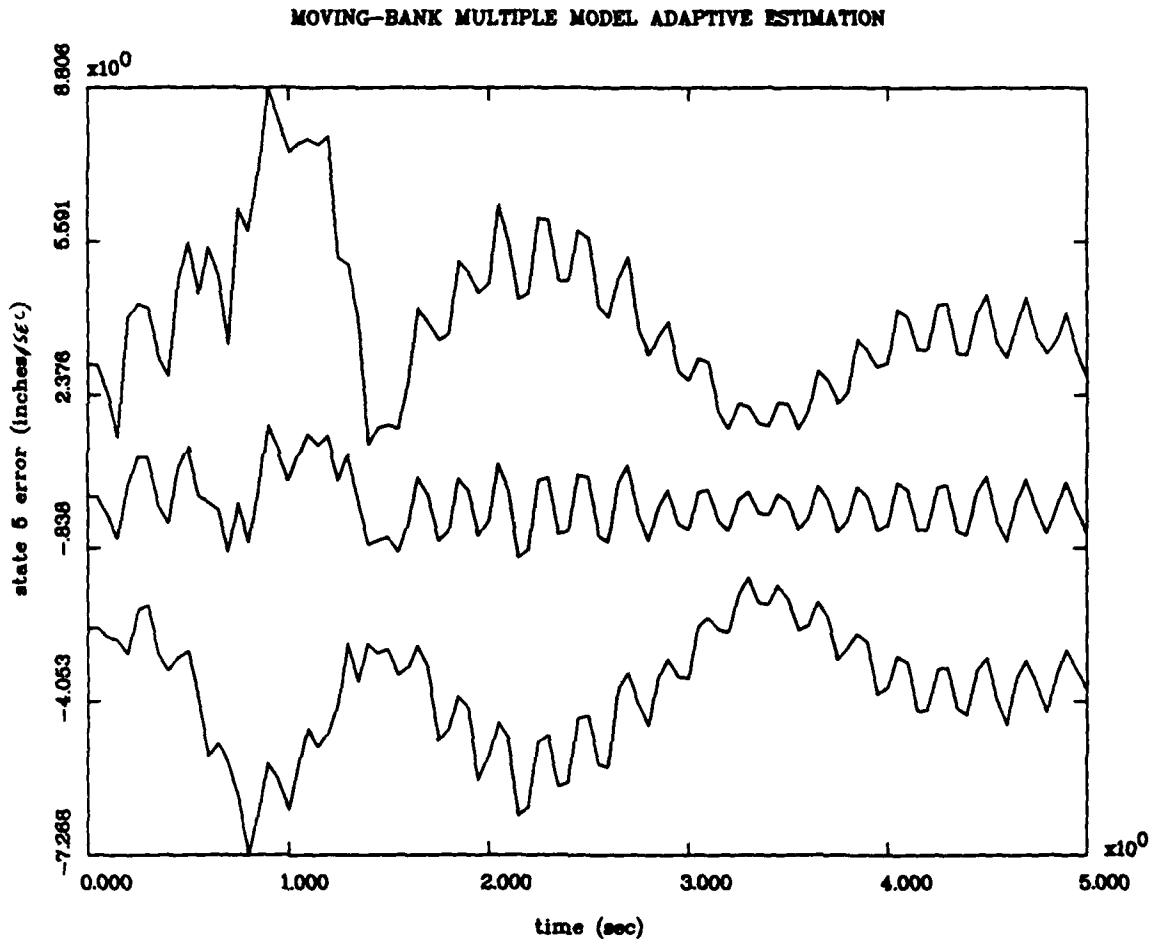

Figure 18(e) State 5
State Estimation Errors
Mean ± 1σ
Noise Level Determination Study
For R2Q5e Case
with 10 Monte Carlo Runs

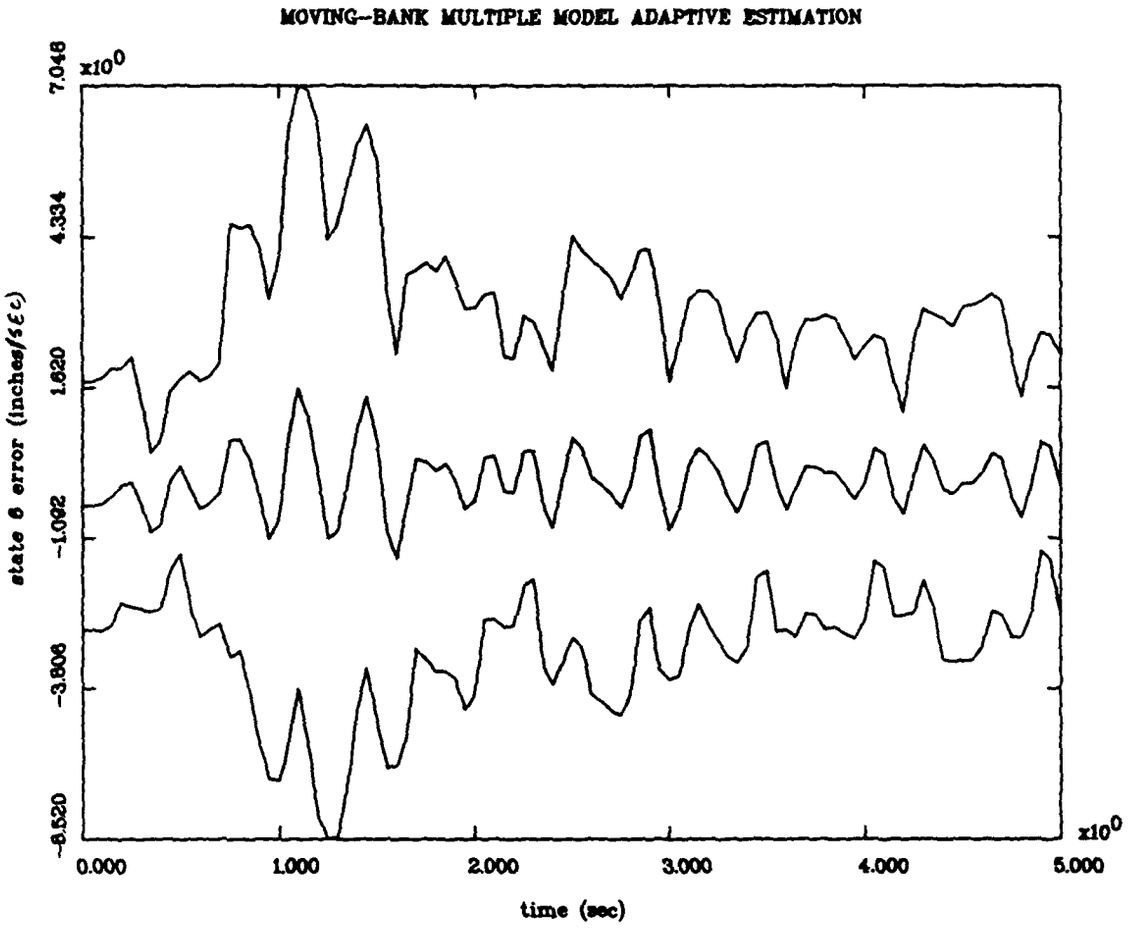

Figure 18(f) State 6
State Estimation Errors
Mean ± 1σ
Noise Level Determination Study
For R2Q5e Case
with 10 Monte Carlo Runs

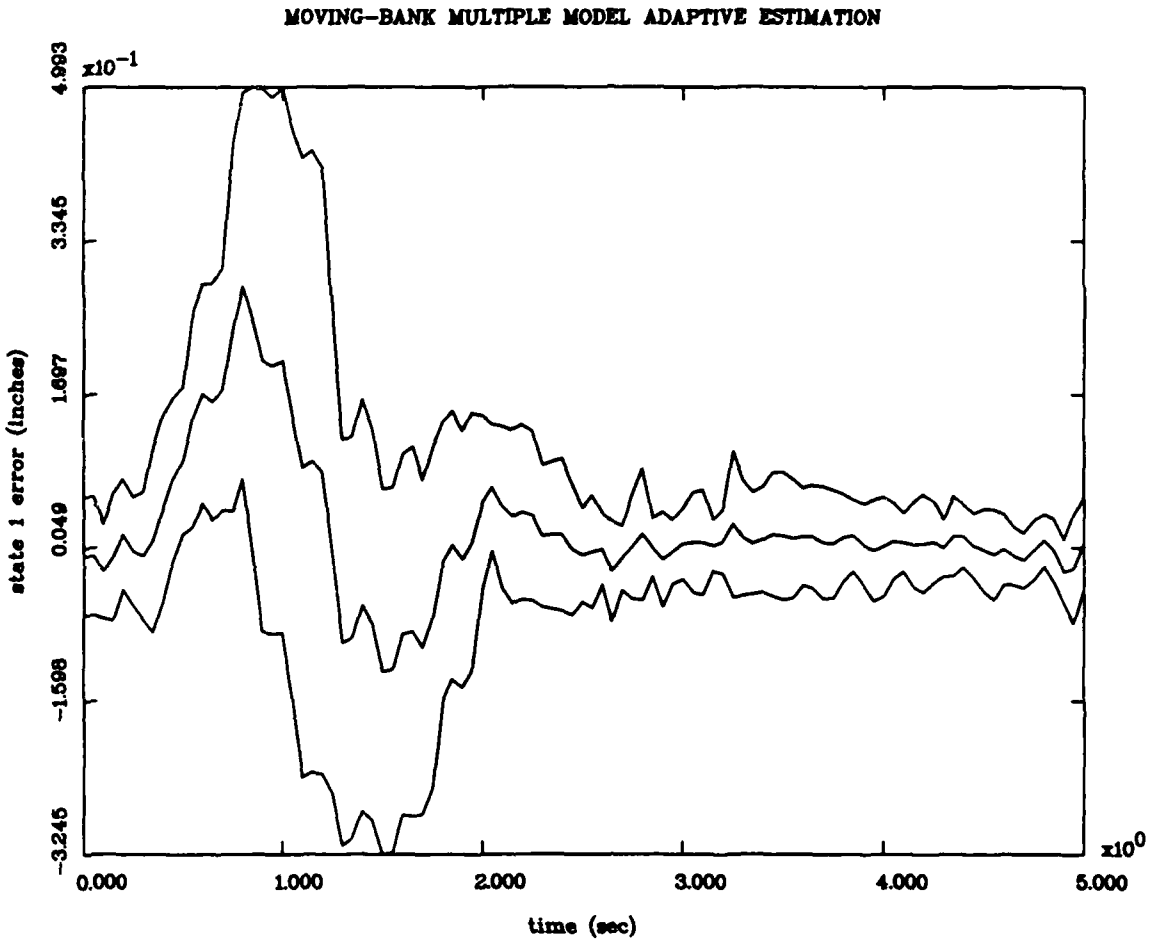

Figure 19(a) State 1
State Estimation Errors
Mean ± 1σ
Noise Level Determination Study
For R2Q10e Case
with 10 Monte Carlo Runs

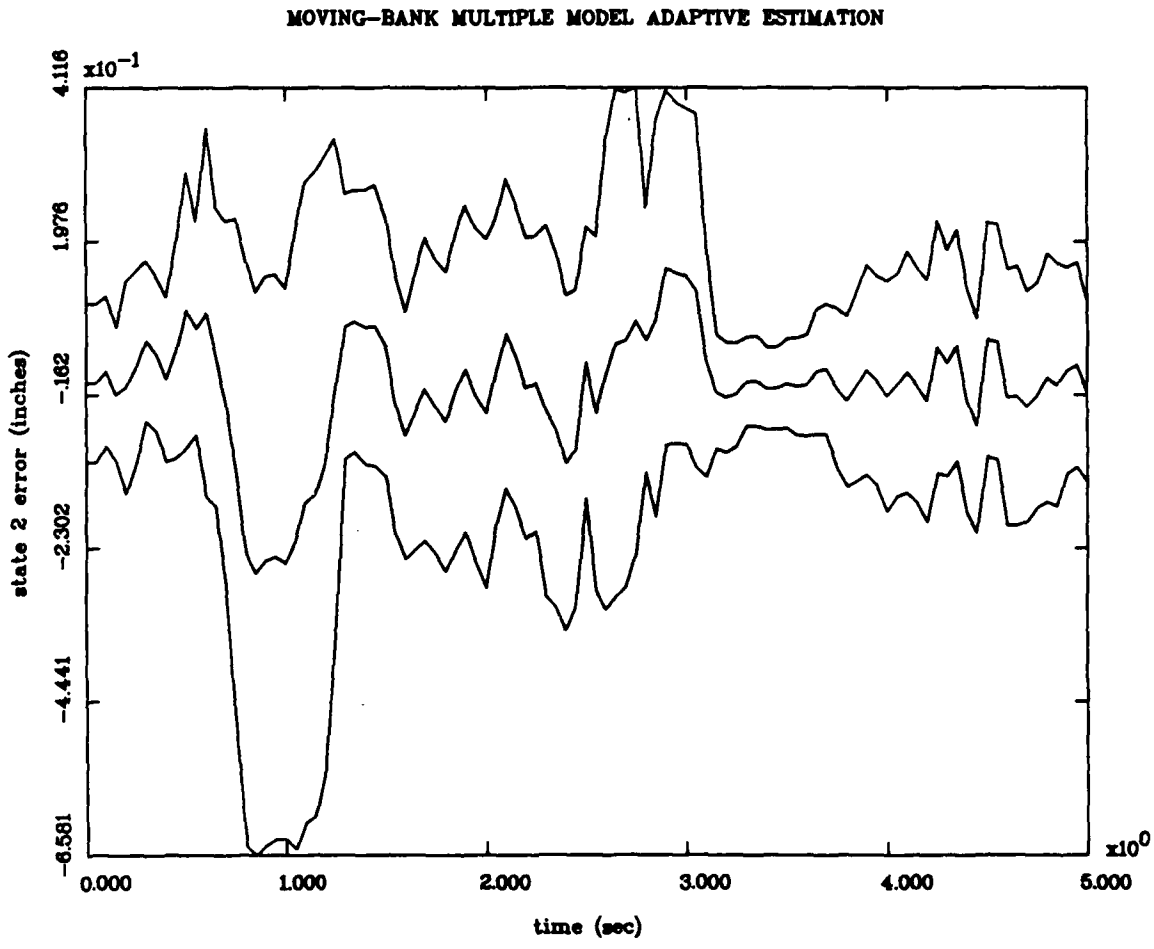

Figure 19(b) State 2
State Estimation Errors
Mean ± 1σ
Noise Level Determination Study
For R2Q10e Case
with 10 Monte Carlo Runs

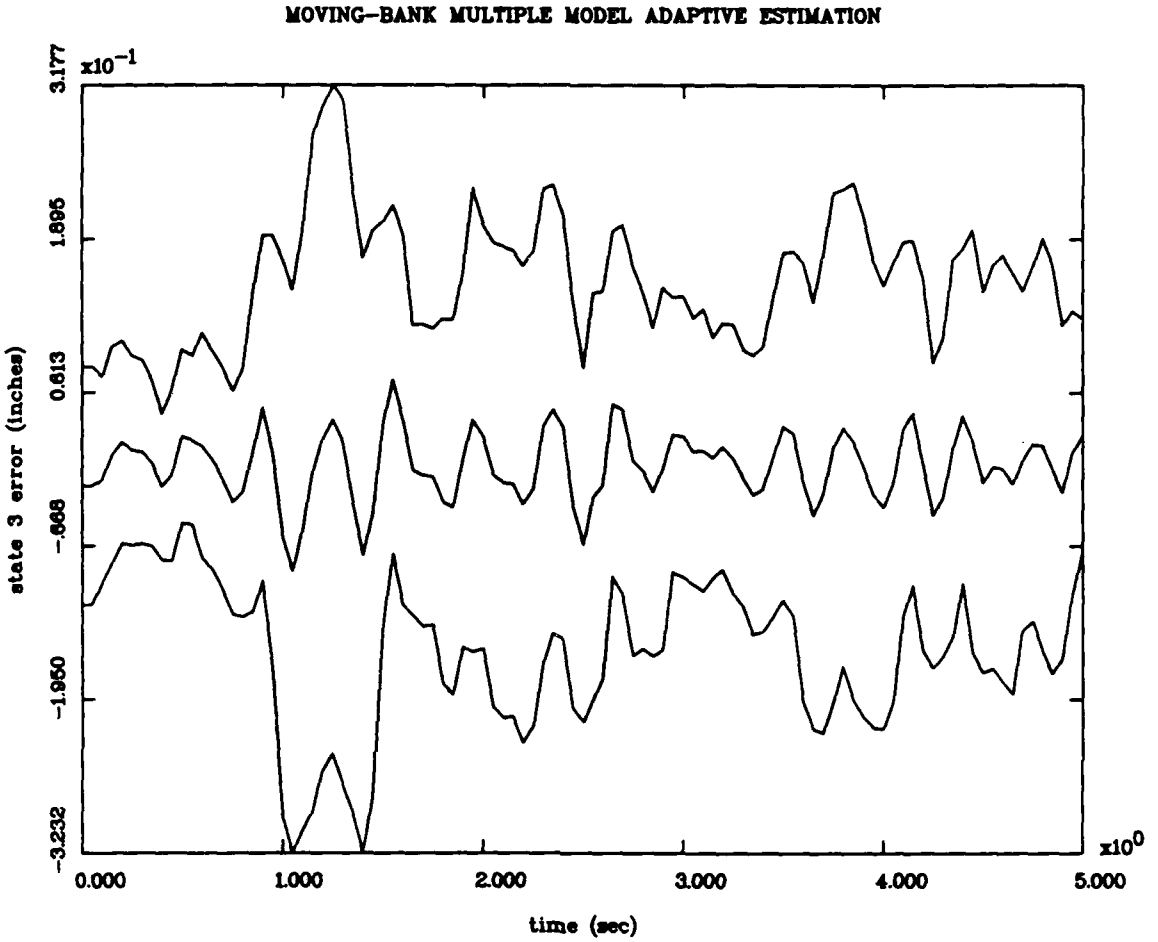

Figure 19(c) State 3
State Estimation Errors
Mean ± 1σ
Noise Level Determination Study
For R2Q10e Case
with 10 Monte Carlo Runs

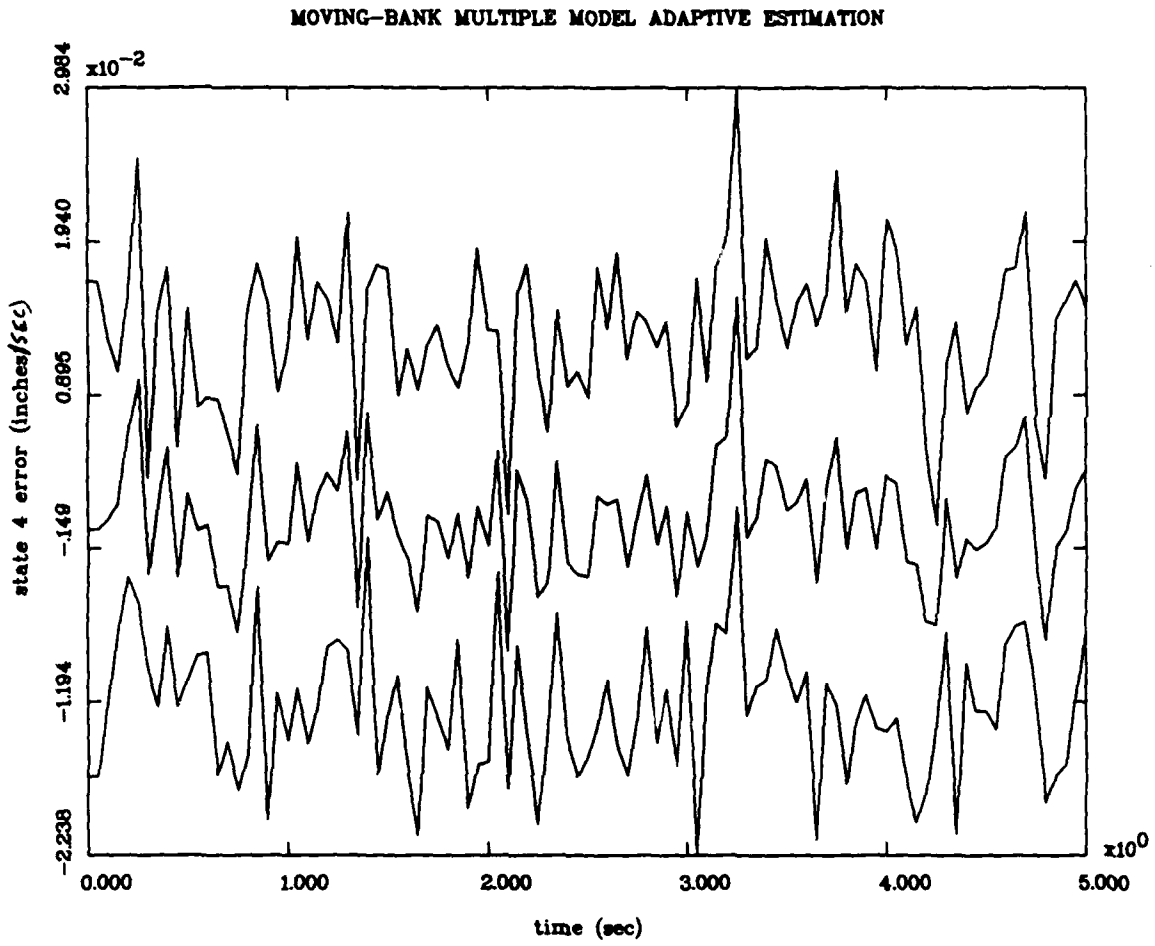

Figure 19(d) State 4
State Estimation Errors
Mean ± 1σ
Noise Level Determination Study
For R2Q10e Case
with 10 Monte Carlo Runs

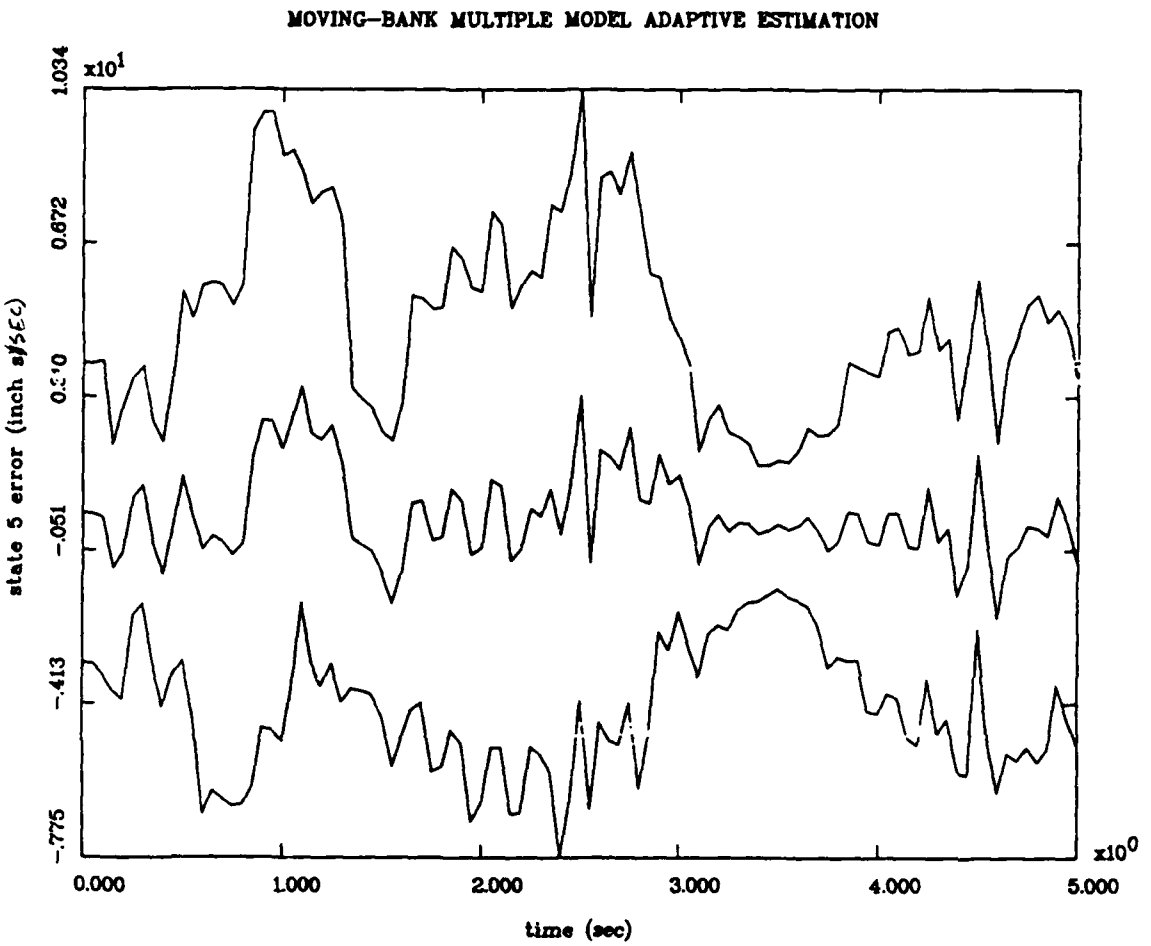

Figure 19(e) State 5
State Estimation Errors
Mean ± 1σ
Noise Level Determination Study
For R2Q10e Case
with 10 Monte Carlo Runs

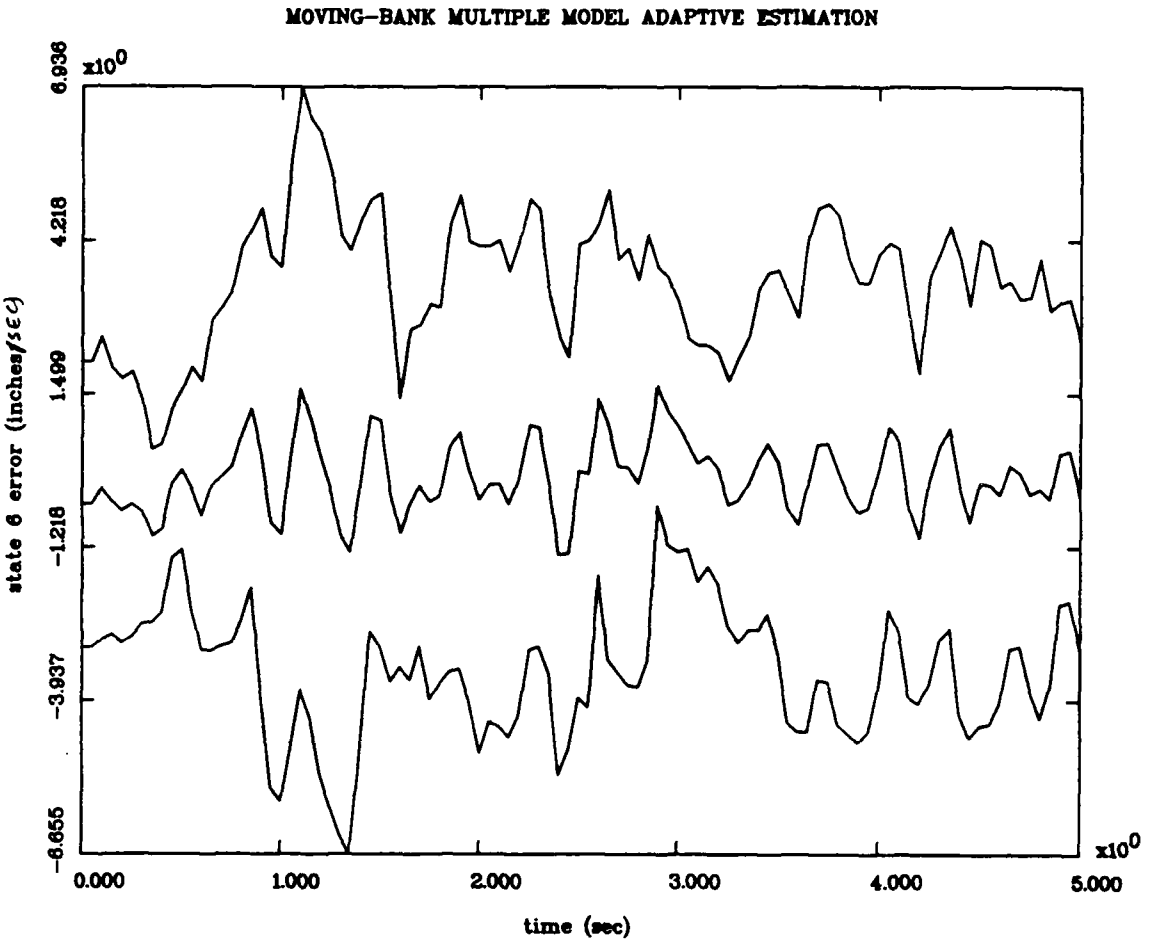

Figure 19(f) State 6
State Estimation Errors
Mean ± 1σ
Noise Level Determination Study
For R2Q10e Case
with 10 Monte Carlo Runs

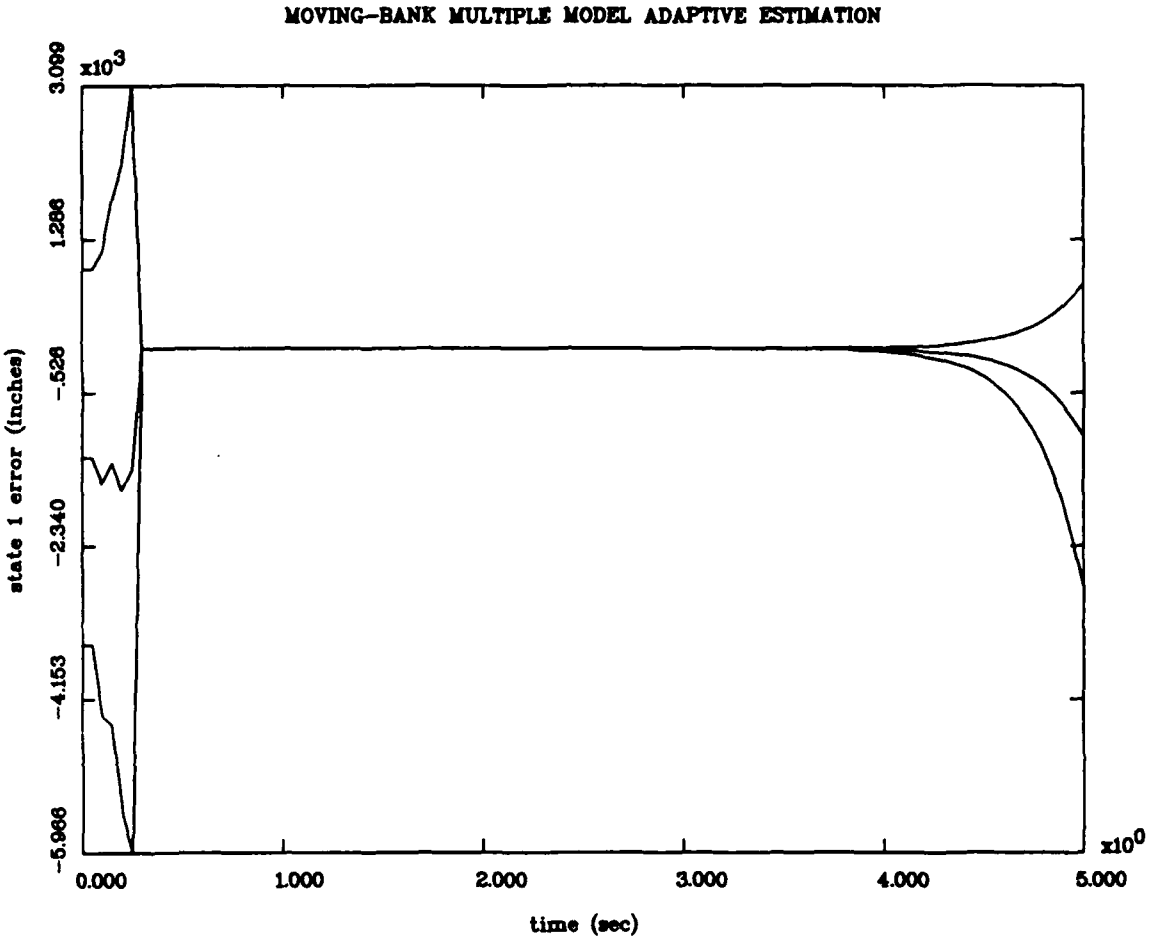

Figure 20(a) State 1
State Estimation Errors
Mean ± 1σ
Noise Level Determination Study
For R3Q10e Case
with 10 Monte Carlo Runs

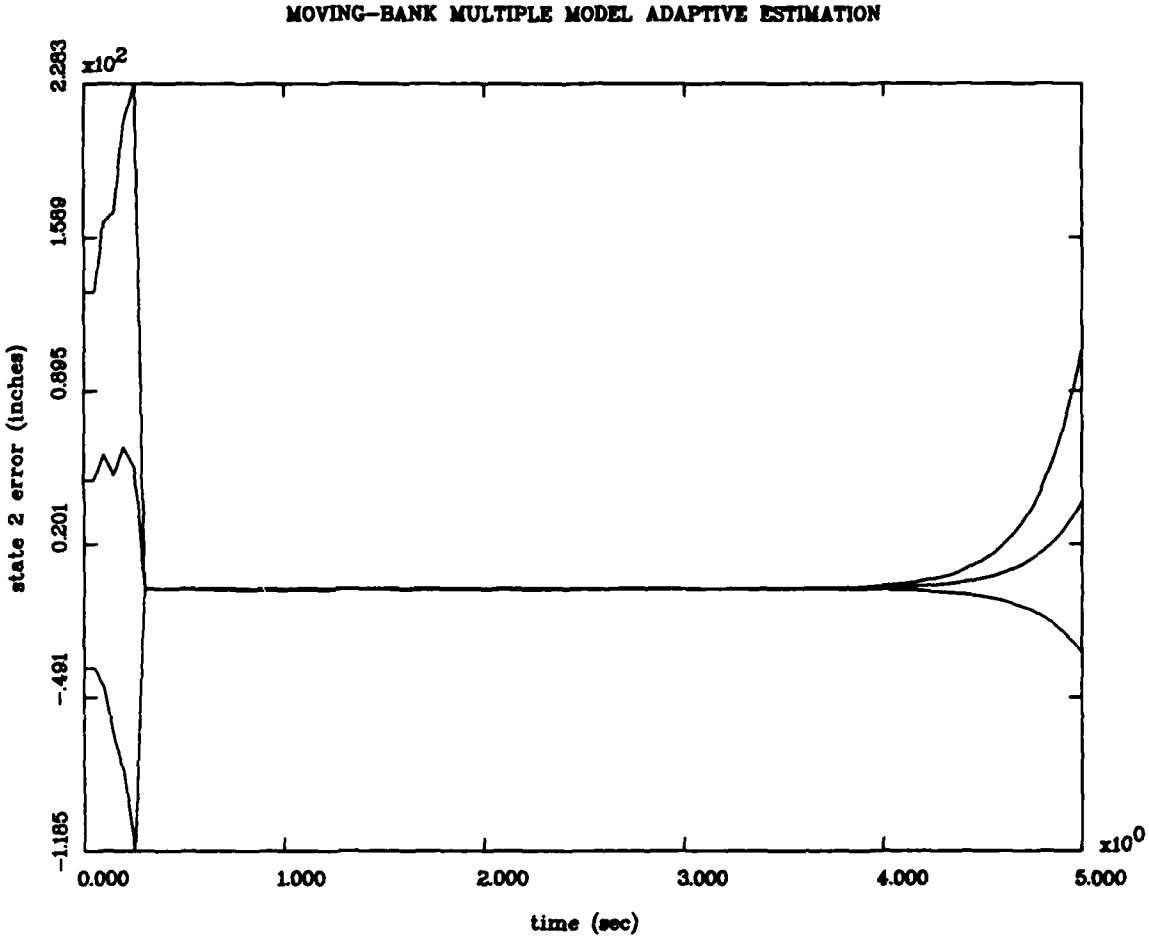

Figure 20(b) State 2
State Estimation Errors
Mean ± 1σ
Noise Level Determination Study
For R3Q10e Case
with 10 Monte Carlo Runs

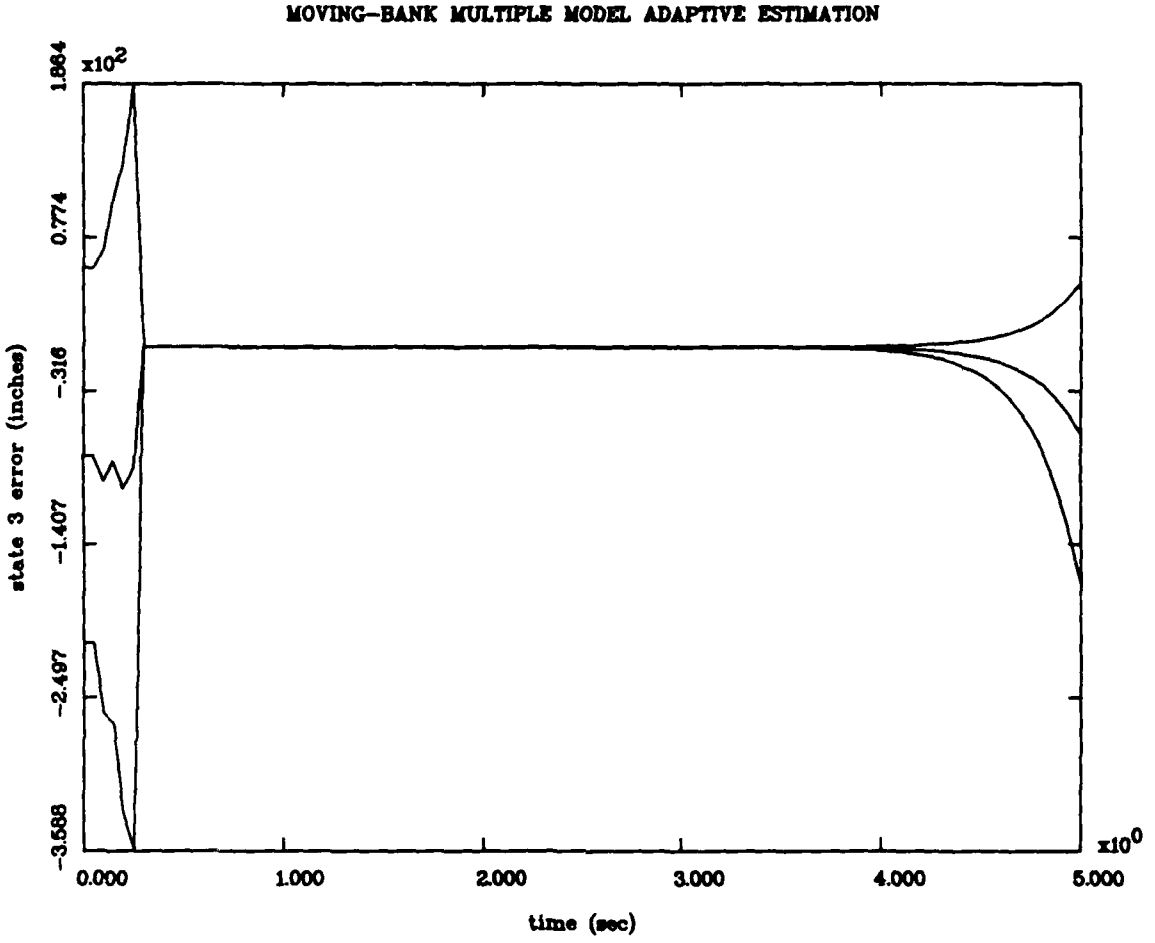

Figure 20(c) State 3
State Estimation Errors
Mean ± 1σ
Noise Level Determination Study
For R3Q10e Case
with 10 Monte Carlo Runs

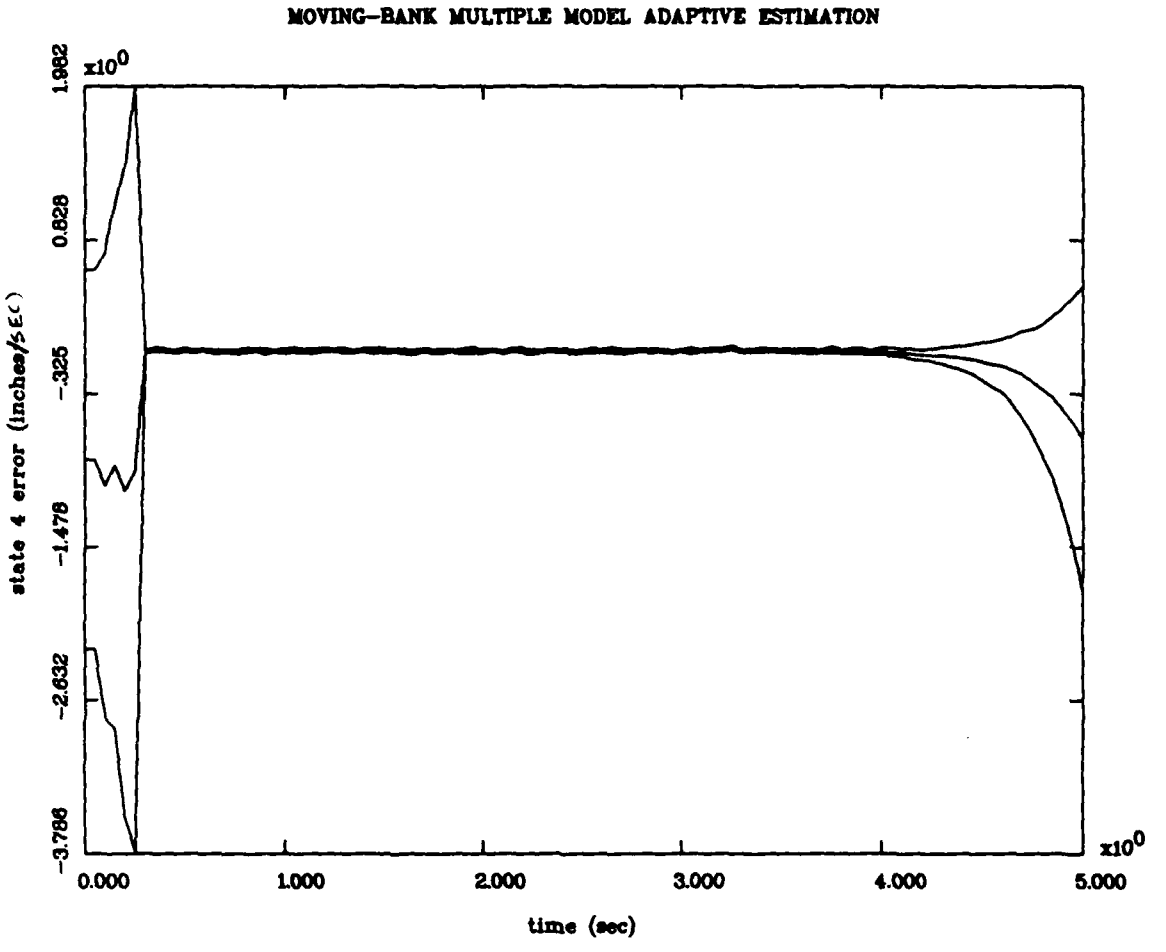

Figure 20(d) State 4
State Estimation Errors
Mean ± 1σ
Noise Level Determination Study
For R3Q10e Case
with 10 Monte Carlo Runs

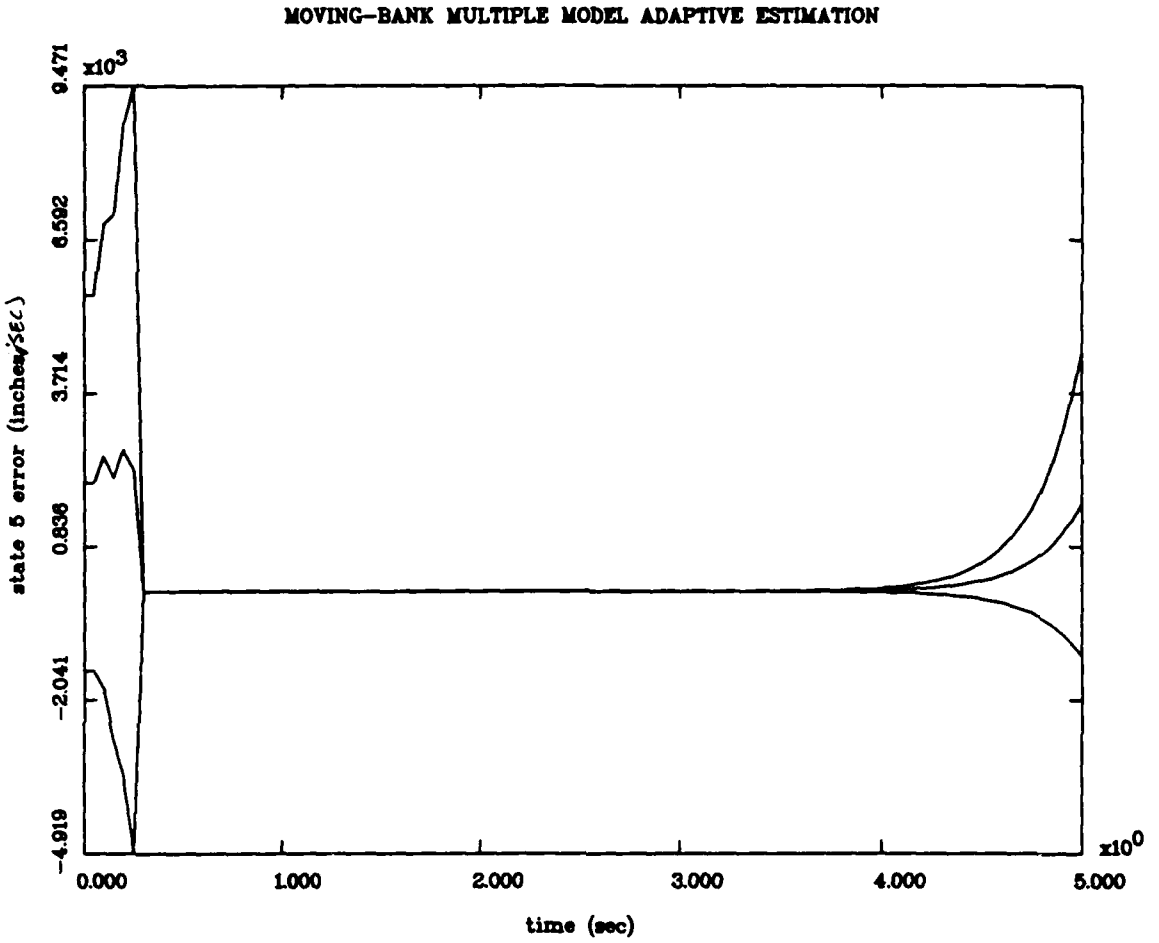

Figure 20(e) State 5
State Estimation Errors
Mean ± 1σ
Noise Level Determination Study
For R3Q10e Case
with 10 Monte Carlo Runs

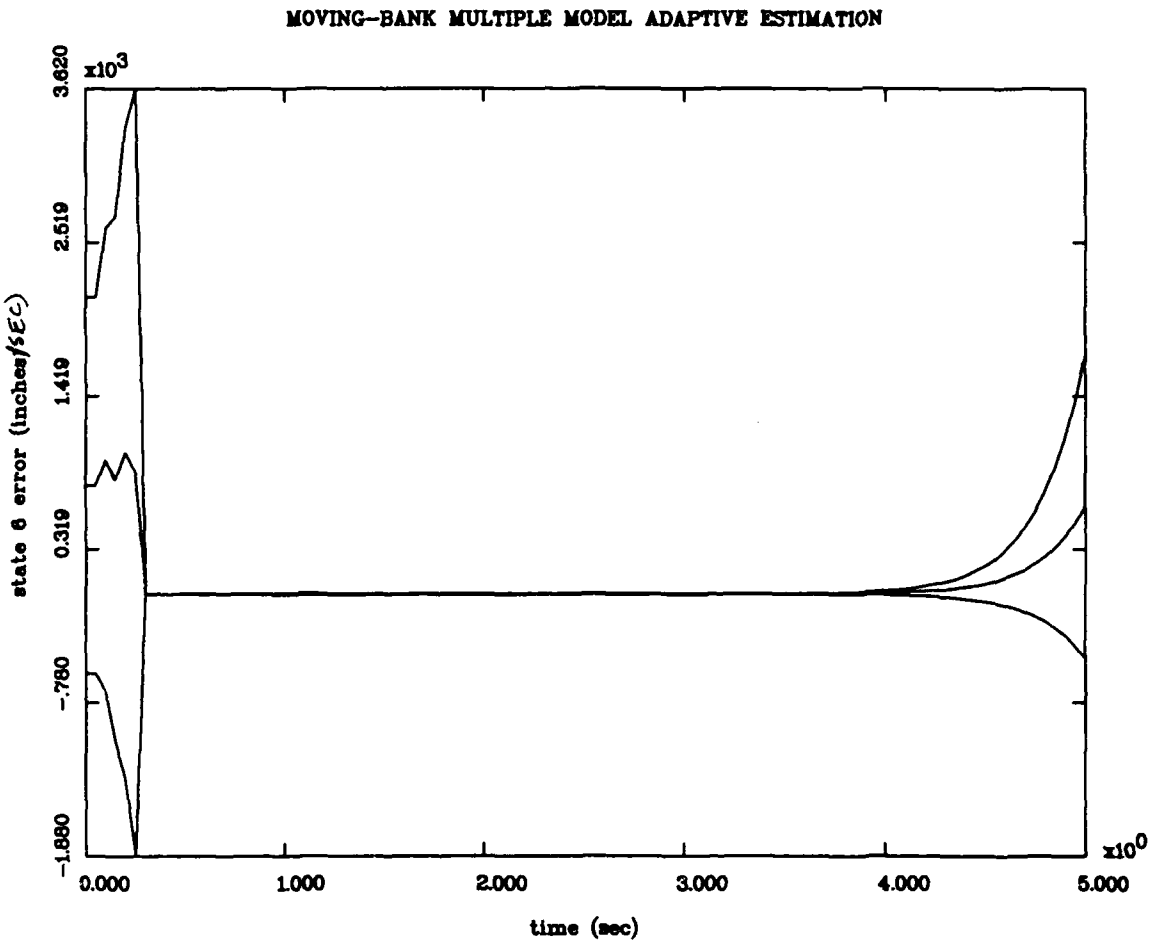

Figure 20(f) State 6
State Estimation Errors
Mean ± 1σ
Noise Level Determination Study
For R3Q10e Case
with 10 Monte Carlo Runs

moving bank to provide accurate parameter estimates. However, the huge difference in the peak values of Figures 15 and 21 suggests that something could be happening during Monte Carlo runs 11-100 that is not evident in the first 10 Monte Carlo runs. This large peak difference (comparing Figures 15 and 21) is similar to the large peak difference discussed previously (comparing Figures 13 and 14), and the probable reasons for this peak difference discussed previously apply here as well. Again, a likely explanation is that the parameter and state estimation process breaks down somehow during the last 90 Monte Carlo runs. Unlike the case previously investigated with multiple peaks in error standard deviation (Figure 14), Figure 21 shows only one peak. This single peak indicates that the moving-bank algorithm produces "better" error state estimates as Q is increased.

Since increasing R_f alone did not provide stable and accurate state estimates, the next logical step was to increase Q. This time $Q = 5 * Q_{old}$ for the R1Q5e case. Figure 16 shows the results of 10 Monte Carlo runs. Clearly, another peak is seen in the first plot; however, only 10 (rather than 100) Monte Carlo runs are required to show this peak, and the height of the peak is at least 6 orders of magnitude lower than the cases (R1Q1e and R1Q2e) previously discussed. Again, this indicates the moving-bank algorithm produces "better" state estimates as Q increases. Again, increasing Q increases both Q_f and Q_t. Better results would probably be generated by increasing Q_f and leaving

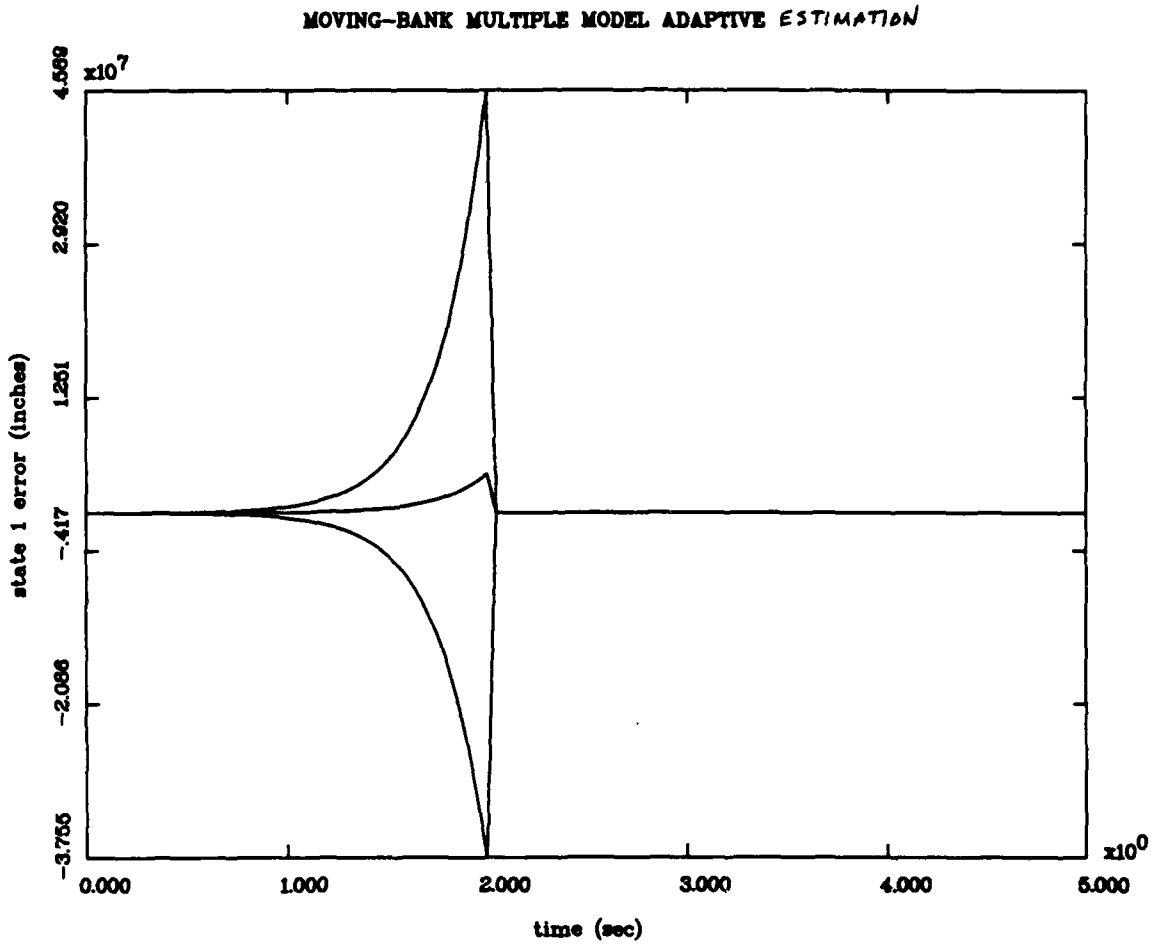

Figure 21(a) State 1
State Estimation Errors
Mean ± 1σ
Noise Level Determination Study
For R1Q2e Case
with 100 Monte Carlo Runs

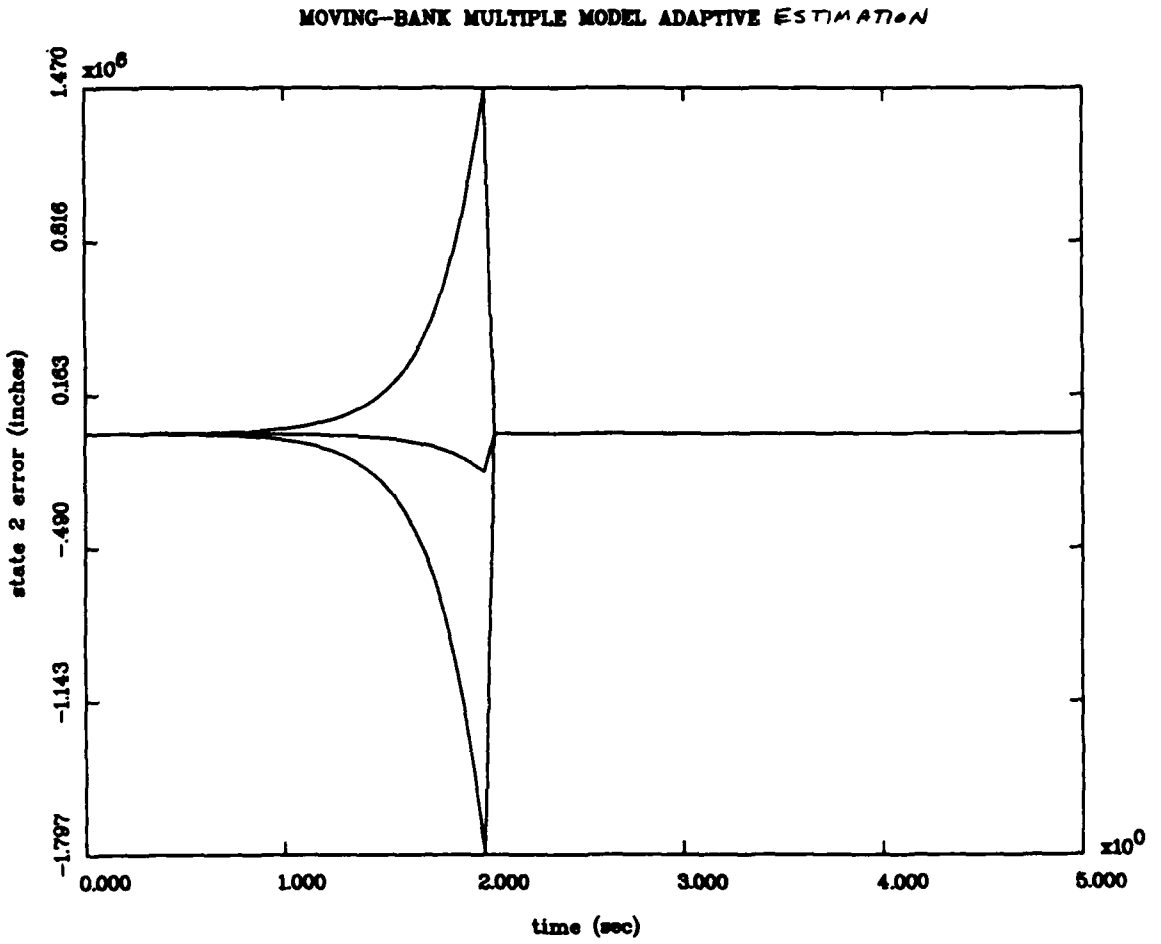

Figure 21(b) State 2
State Estimation Errors
Mean ± 1σ
Noise Level Determination Study
For R1Q2e Case
with 100 Monte Carlo Runs

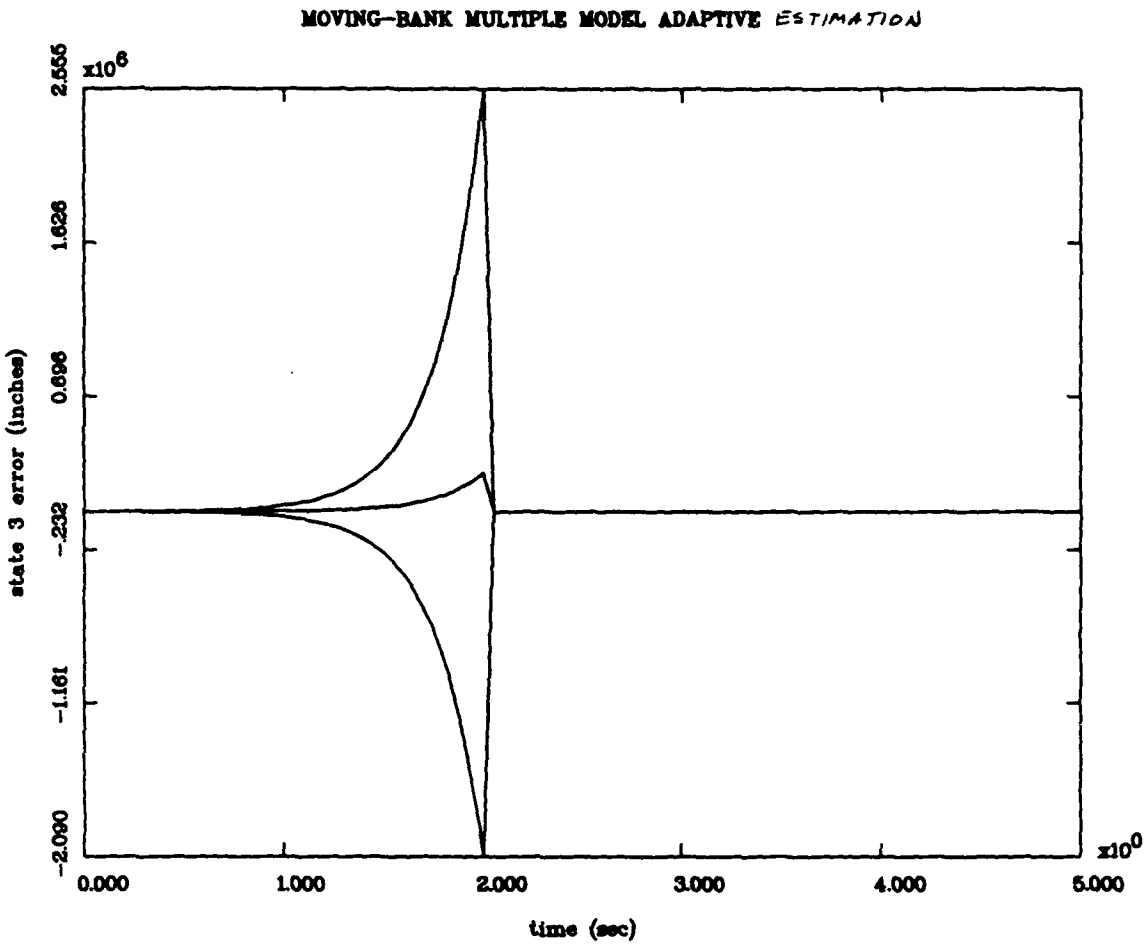

Figure 21(c) State 3
State Estimation Errors
Mean ± 1σ
Noise Level Determination Study
For R1Q2e Case
with 100 Monte Carlo Runs

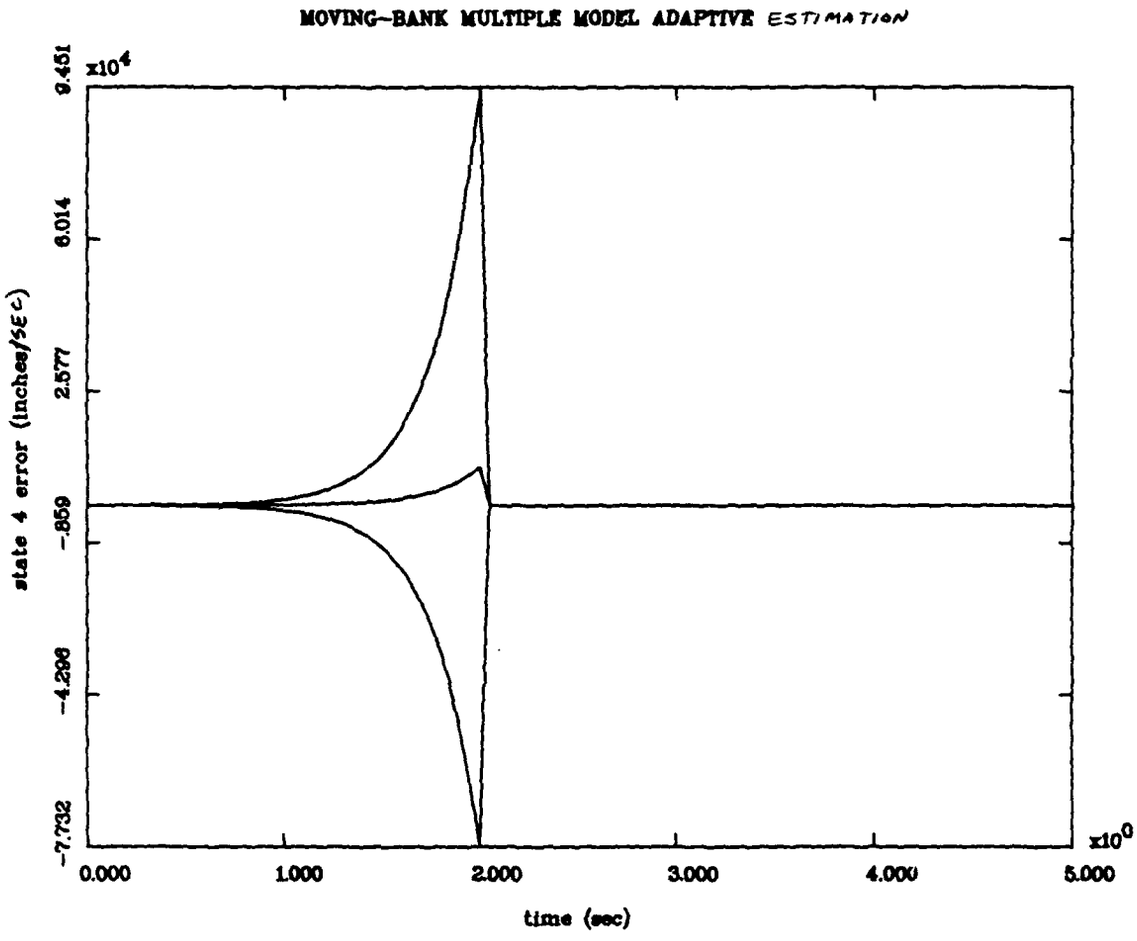

Figure 21(d) State 4
State Estimation Errors
Mean ± 1σ
Noise Level Determination Study
For R1Q2e Case
with 100 Monte Carlo Runs

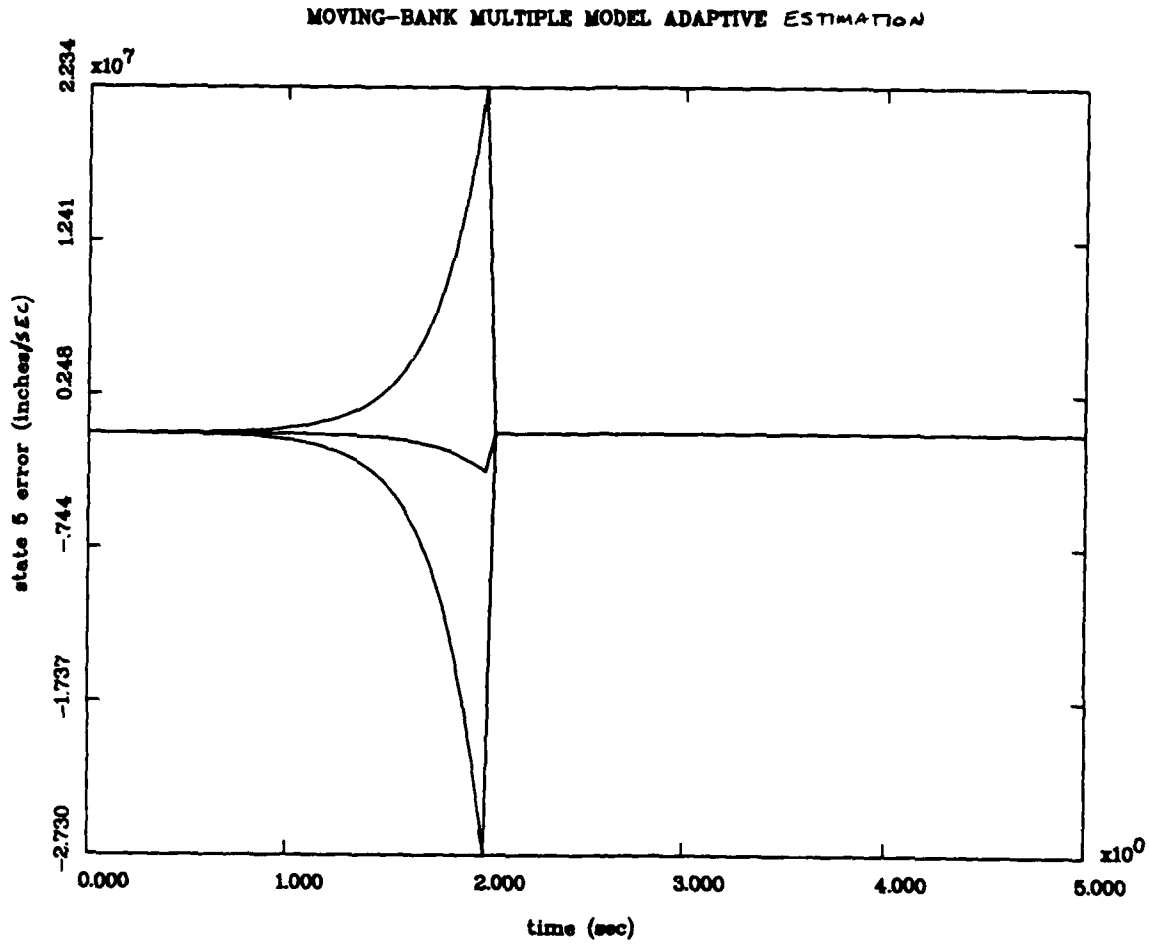

Figure 21(e) State 5
State Estimation Errors
Mean ± 1σ
Noise Level Determination Study
For R1Q2e Case
with 100 Monte Carlo Runs

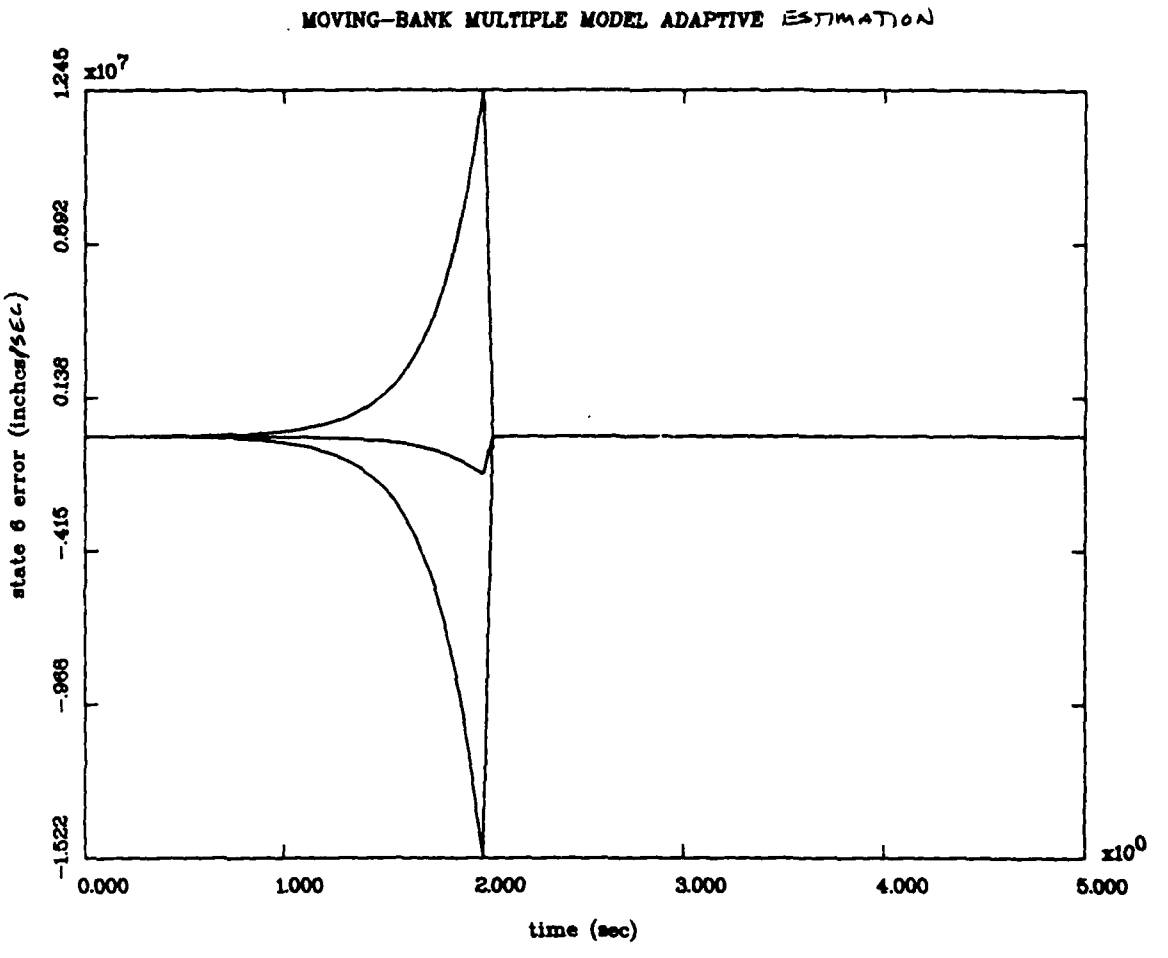

Figure 21(f) State 6
State Estimation Errors
Mean ± 1σ
Noise Level Determination Study
For R1Q2e Case
with 100 Monte Carlo Runs

Q_t as originally set by Lashlee [8:94]. However, this true filter tuning was not performed in this thesis effort. The results from this case indicate that the algorithm is still not able to provide accurate parameter estimates.

A set of time history plots of the mean plus or minus one standard deviation for the parameter estimates would be both helpful and instructive here. Unfortunately, the software was unable to generate such plots during this thesis research. The software did, however, generate a listing of the parameters and the number of times the moving bank provided estimates at those parameters (but not when in the simulation these estimates were generated). Table 5 lists the parameters the center of the moving bank used and the number of times the center is located at that point in the parameter space. The unused parameters are not listed here. Note that for the duplication of past work effort, described in Sections 4.4.1 and 5.2.1, the moving bank was able to "lock" on to the correct parameters for all cases tested (even when the bank was initially centered at an incorrect parameter value). This indicates that accurate parameter estimation is possible, but it is most likely totally confounded by the effects of the unmodeled states.

Table 5
Parameters Used and the Number of Times Used
During the Monte Carlo Simulation for the Case R2Q5e
The True Parameter Value is (7,6)
(Note: this is not a time history)

Parameter Value	Number of Times Used
(7,5)	10
(7,6)	3
(8,5)	2
(9,4)	896

As demonstrated by Table 5, the moving bank only used the true parameter (7,6) three times to provide estimates. The table also shows a slight "drifting" of the moving bank around the parameter space. Even though the time history is not available, the fact that the bank used the parameter (9,4) 896 times demonstrates that it did "lock" on to this value at some time during the simulation.

The result of increasing Q to $Q = 10 * Q_{old}$ (R1Q10e case) is shown in Figure 17. These plots show no peaks. These error state estimates appear stable and accurate. Even though the algorithm is still not able to provide accurate parameter estimates, accurate state estimates are produced. Raising the value of Q above $Q = 10 * Q_{old}$ has proven to be counterproductive. Neither accurate state nor accurate parameter estimates are produced when larger values of Q are used.

Since larger Q values were unproductive, it was decided to try increasing the R_f matrix and investigate these results. Even

though increasing the R_f matrix tells the filter to "distrust" the incoming measurements more and thereby rely more on it internal model, better state estimation results were generated when R_f was increased. Even better results would probably have been found by increasing the Q_f matrix (and leaving the Q_t alone), but this Q_f tuning was not done for this study.

Raising R_f to $R_f = 2 * R_{f\text{-}old}$ and setting Q to $Q = 5 * Q_{old}$ (R2Q5e case) produces the results shown in Figure 18. These plots indicate an increase in the error state estimation accuracy over the cases previously used. However, the algorithm is still unable to lock on to accurate parameter estimates.

Leaving R_f the same and increasing Q to $Q = 10 * Q_{old}$ (R2Q10e case) produces the results shown in Figure 19. Figures 18 and 19 both indicate stable and accurate error state estimations.

Finally, the case of R3Q10e is used. Figure 20 shows the results of this case. Obviously, the result is unstable. This case demonstrates that the R_f and Q values may be set only so high before the MMAE algorithm produces unstable results. The large value of Q_t which results from increasing Q_f and Q_t simultaneously probably causes a great deal of the instabilities and inaccuracies.

The objective of this noise determination study is to find R_f and Q values which provide stable and accurate estimates of the states. This objective is accomplished by some of the cases listed previously (namely R1Q10e, R2Q5e, and R2Q10e). The other cases are either unstable (R3Q10e) or have extremely large peaks in the standard deviation (R1Q1e, R1Q2e, and R1Q5e) which are undesirable. For some of the plots, a pseudo-sinusoidal rise and fall in the error state estimates can be seen. This rise and fall may be explained by a "beat frequency" which results when the true parameter differs from the estimated parameter. This "beat frequency" results when the actual motion of the two-bay truss differs from the "expected" motion generated by the filter estimates. This phenomenon can also be attributed to the higher order bending mode effects that are not modeled by the filter. For all the cases previously listed, the movable bank of Kalman filters is initially centered at the true parameter (7,6).

The effect of increasing the R_f matrix "tells" the filter that its representation of the real world is in error, specifically due to the unmodeled contributions by higher order bending modes to the three positions and three velocities of interest. However, simply increasing R_f alone is not enough to compensate for the mismodeling because this R_f increase in effect decreases the overall Kalman filter gain. Therefore, the Q_f matrix must be increased as well. The next section discusses the robustness issue and what happens when the movable bank of

filters is not initially centered at the true parameter. Throughout this noise determination study, no values of the R_f and Q matrices have been found which provide accurate parameter estimates. In fact, the parameter estimate either wanders around the parameter space or locks on to a parameter which is not the true parameter.

5.2.3 Robustness Analysis

The robustness analysis is performed as described in Sections 1.4 and 4.4.3. As previously stated, the robustness analysis is the main goal of this thesis research. Basically, three cases are used in the robustness analysis of the mismatched (24-state truth model and 6-state filter model) moving-bank MMAE algorithm. The first case involves running the Monte Carlo simulation with a single 6-state non-adaptive filter, set at the wrong parameter (5,5), against the 24-state truth model. The second case involves running the Monte Carlo simulation with a single 6-state non-adaptive artificially informed filter, set at the true parameter (7,6), against the 24-state truth model. Finally, the third case involves running the Monte Carlo simulation with the adaptive moving-bank of 6-state filters, started at the true parameter (7,6) and at the incorrect parameter (5,5), against the 24-state truth model. The R_f and Q matrices are not retuned in this section. They remain at values found from the previous section which provide stable and accurate state estimates.

The results of this first case are shown in the mean plus or minus one standard deviation plots of Figure 22. Note that the dither signal is "turned off" at 0.5 second. As shown by this figure, the state estimate errors are unstable. This instability is expected since the parameters provided the bank of filters are wrong. The single filter is unable to adapt by moving about the parameter space, so the state estimate are incorrect and unstable.

The results for the second case are shown in the mean plus or minus one standard deviation plots of Figure 23. As shown by these plots, the state estimate errors are stable, but the plotted values tend to rise and fall in a pseudo-sinusoidal manner. The parameter "estimates" are always artificially forced to the true parameter. This pseudo-sinusoidal rise and fall in the error state estimate is most likely due to the unmodeled effects of the higher bending modes.

The results of the third case has already been covered by the noise level determination study when the movable bank is centered at the true parameter (7,6). As the noise level determination study demonstrates, the unmodeled states do indeed affect the state estimation, but increasing the Q and R_f matrices can compensate for the effects. However, in direct opposition to Lashlee's results for a 6-state truth model [8:115; 9:21], the

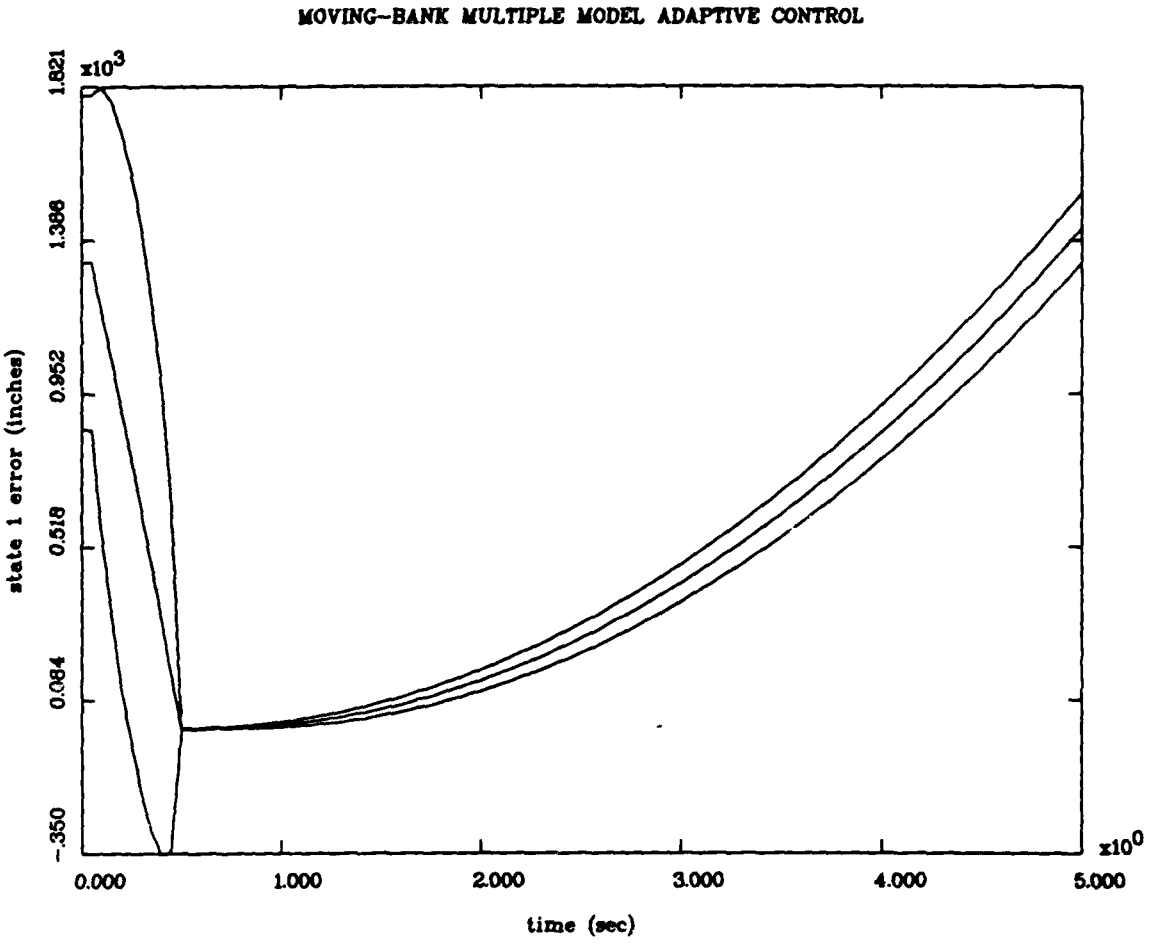

Figure 22(a) State 1
State Estimation Errors
Mean ± 1σ
Robustness Study
For Single Non-Adaptive Uninformed Filter
with 10 Monte Carlo Runs

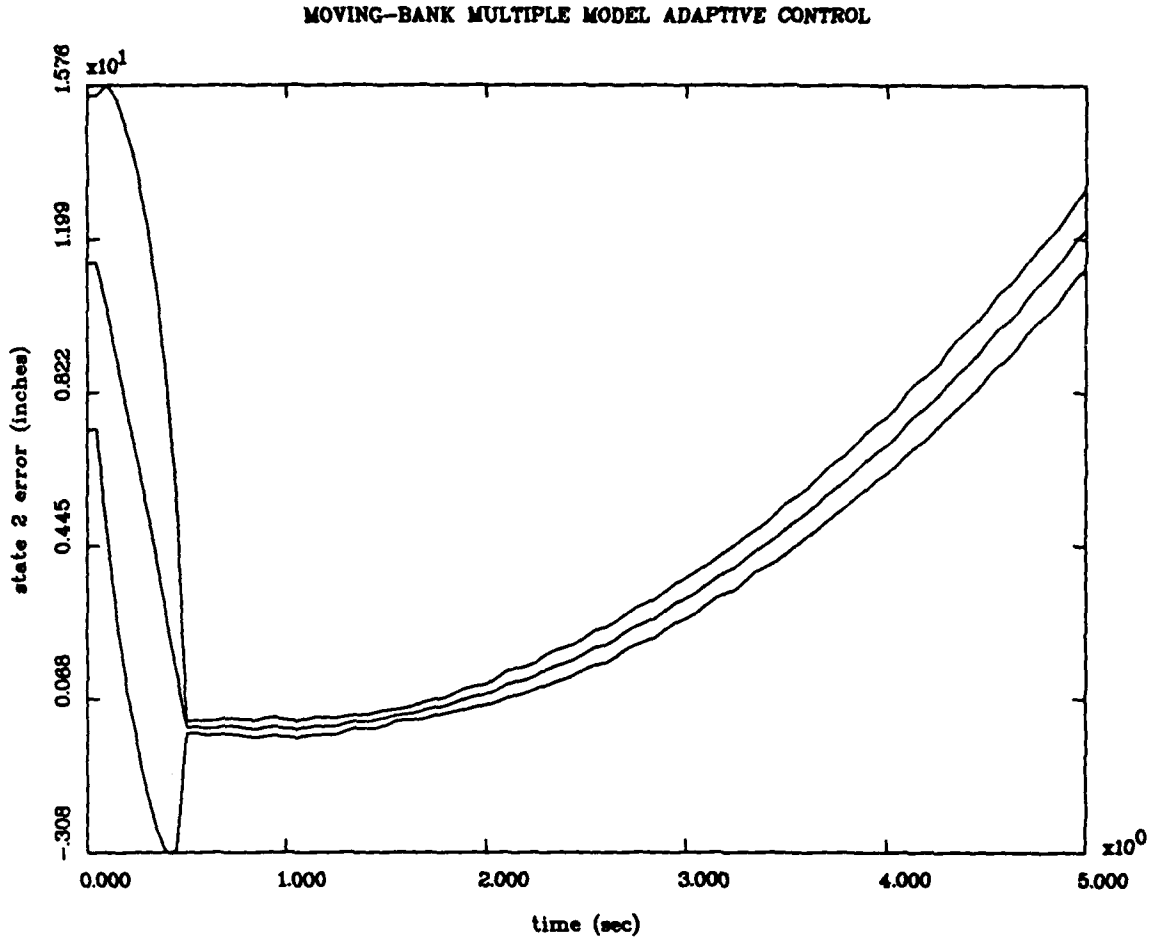

Figure 22(b) State 2
State Estimation Errors
Mean ± 1σ
Robustness Study
For Single Non-Adaptive Uninformed Filter
with 10 Monte Carlo Runs

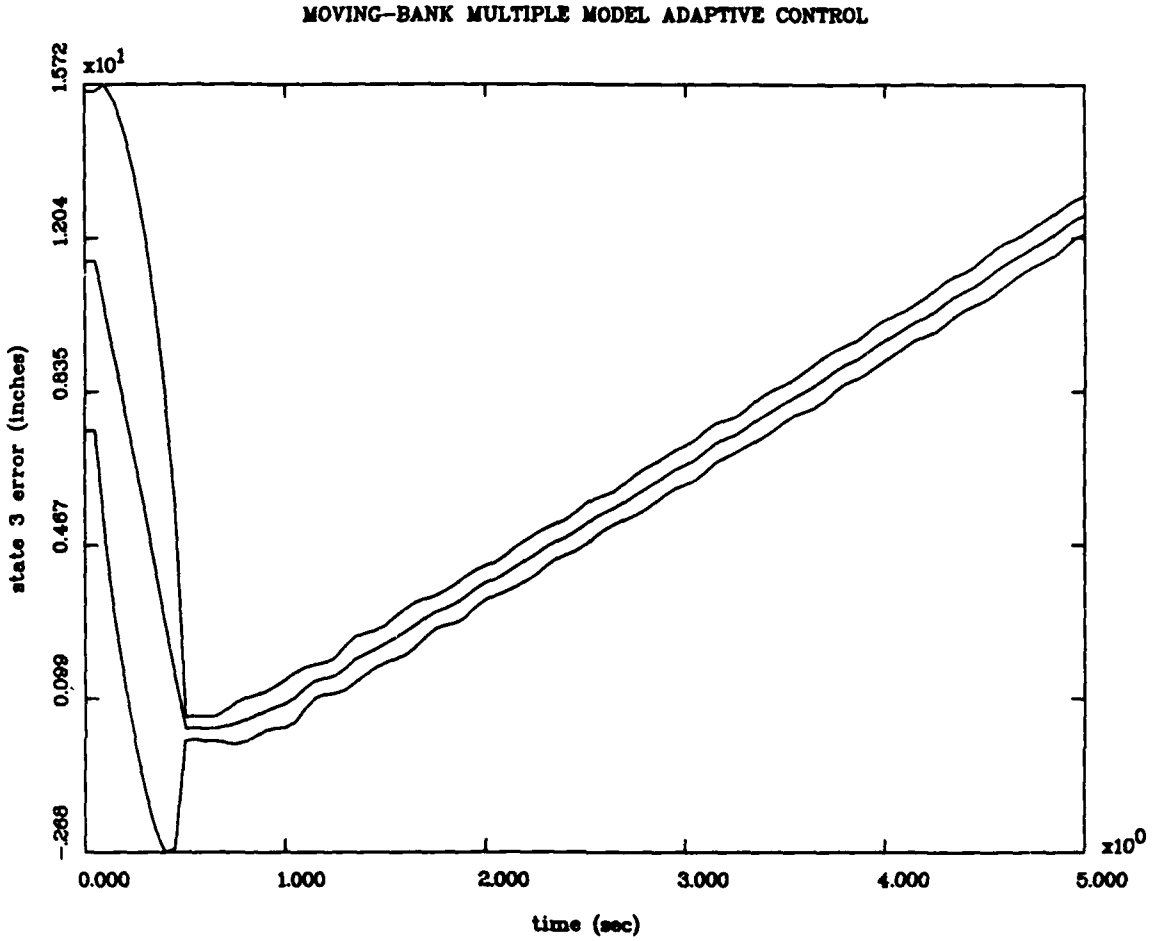

Figure 22(c) State 3
State Estimation Errors
Mean ± 1σ
Robustness Study
For Single Non-Adaptive Uninformed Filter
with 10 Monte Carlo Runs

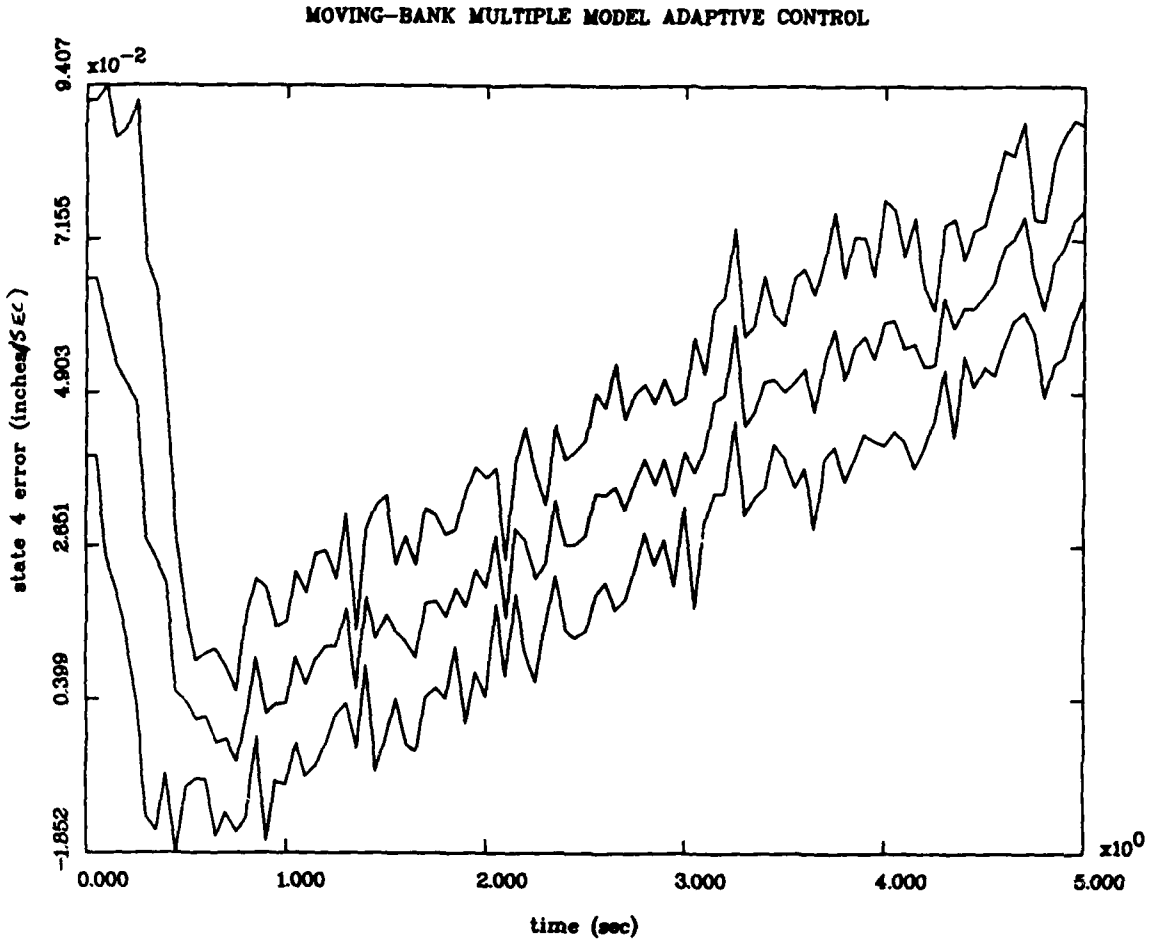

Figure 22(d) State 4
State Estimation Errors
Mean ± 1σ
Robustness Study
For Single Non-Adaptive Uninformed Filter
with 10 Monte Carlo Runs

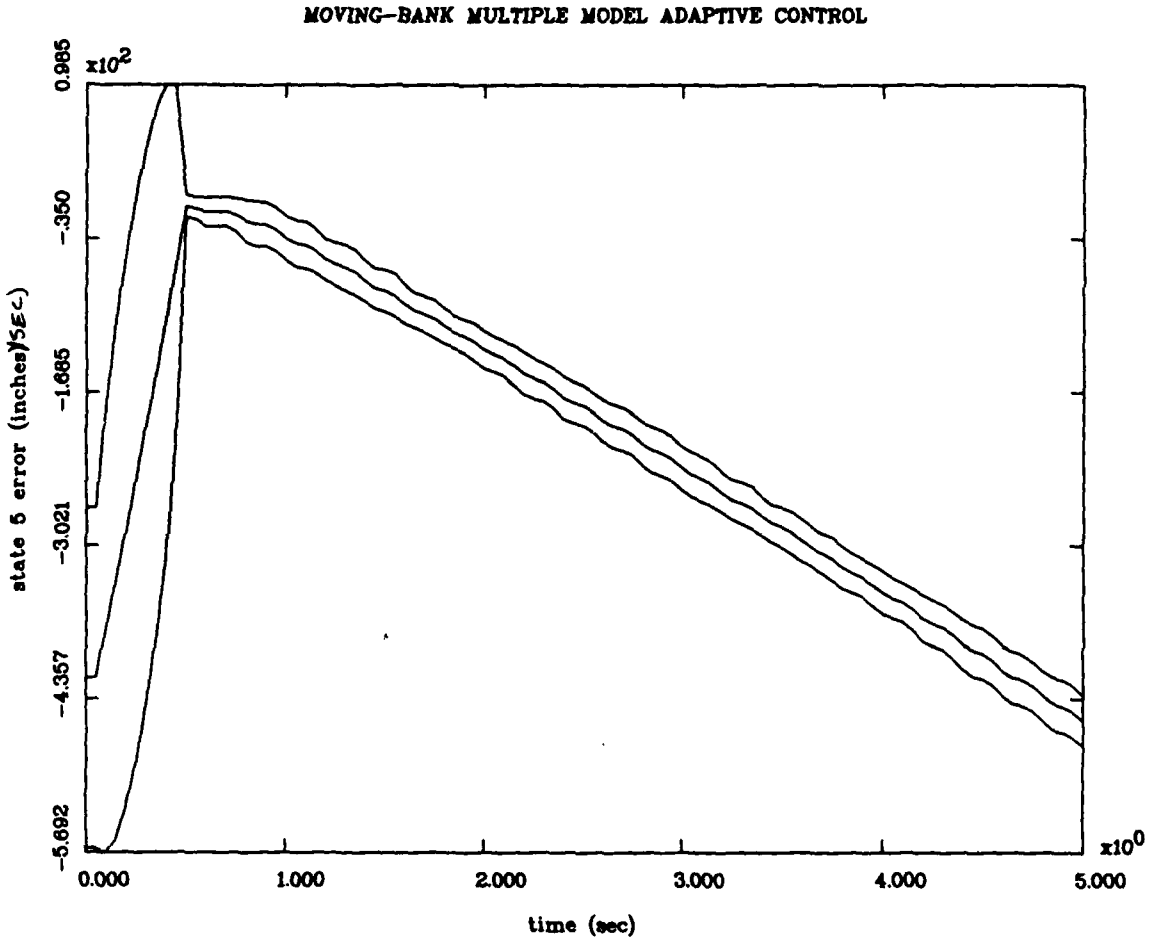

Figure 22(e) State 5
State Estimation Errors
Mean ± 1σ
Robustness Study
For Single Non-Adaptive Uninformed Filter
with 10 Monte Carlo Runs

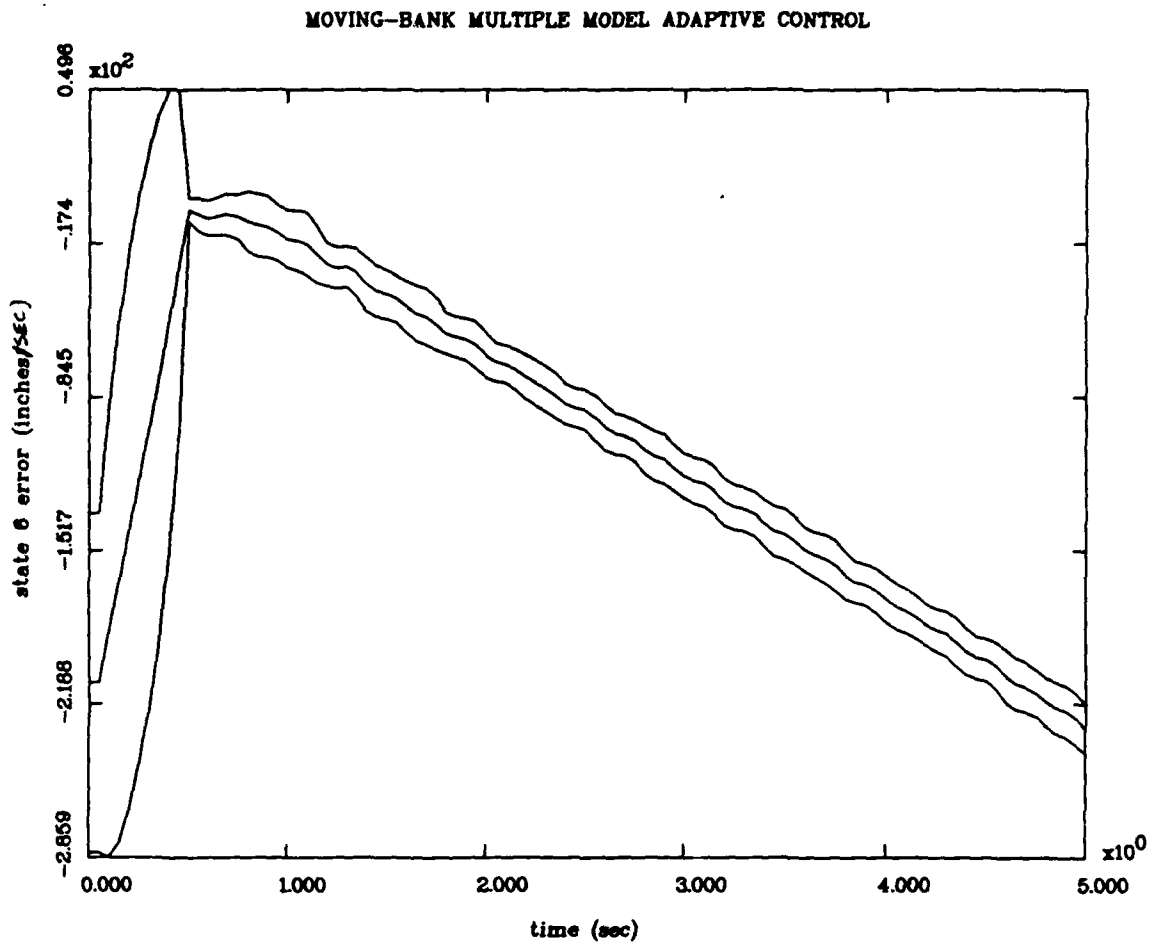

Figure 22(f) State 6
State Estimation Errors
Mean ± 1σ
Robustness Study
For Single Non-Adaptive Uninformed Filter
with 10 Monte Carlo Runs

by moving about the parameter space, so the state estimate are incorrect and unstable.

The results for the second case are shown in the mean plus or minus one standard deviation plots of Figure 23. As shown by by these plots, the state estimate errors are stable, but the plotted values tend to rise and fall in a pseudo-sinusoidal manner. The parameter "estimates" are always artificially forced to the true parameter. This pseudo-sinusoidal rise and fall in the error state estimate is most likely due to the unmodeled effects of the higher bending modes.

The results of the third case has already been covered by the noise level determination study when the movable bank is centered at the true parameter (7,6). As the noise level determination study demonstrates, the unmodeled states do indeed affect the state estimation, but increasing the Q and R_f matrices can compensate for the effects. However, in direct opposition to Lashlee's results for a 6-state truth model [8:115; 9:21], the movable bank cannot provide accurate parameter estimates. This inability to provide accurate parameter estimates <u>may</u> be due to the fact that Q_t increased. Likewise, Lashlee concludes that:

> ...there is a strong correlation between the ability of the moving-bank algorithm to estimate the true parameter and its ability to provide precise state estimates. [9:21]

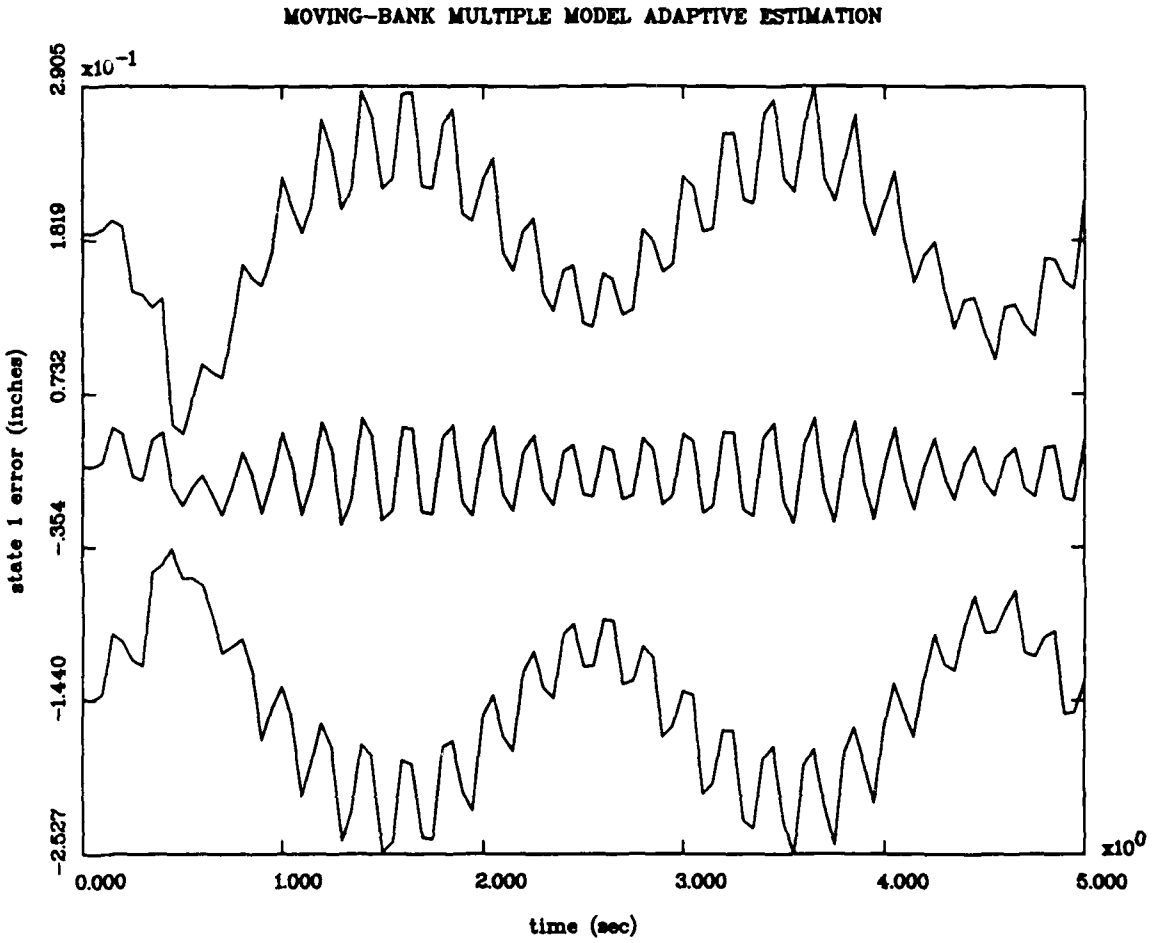

Figure 23(a) State 1
State Estimation Errors
Mean ± 1σ
Robustness Study
For Single Non-Adaptive Artificially Informed Filter
with 10 Monte Carlo Runs

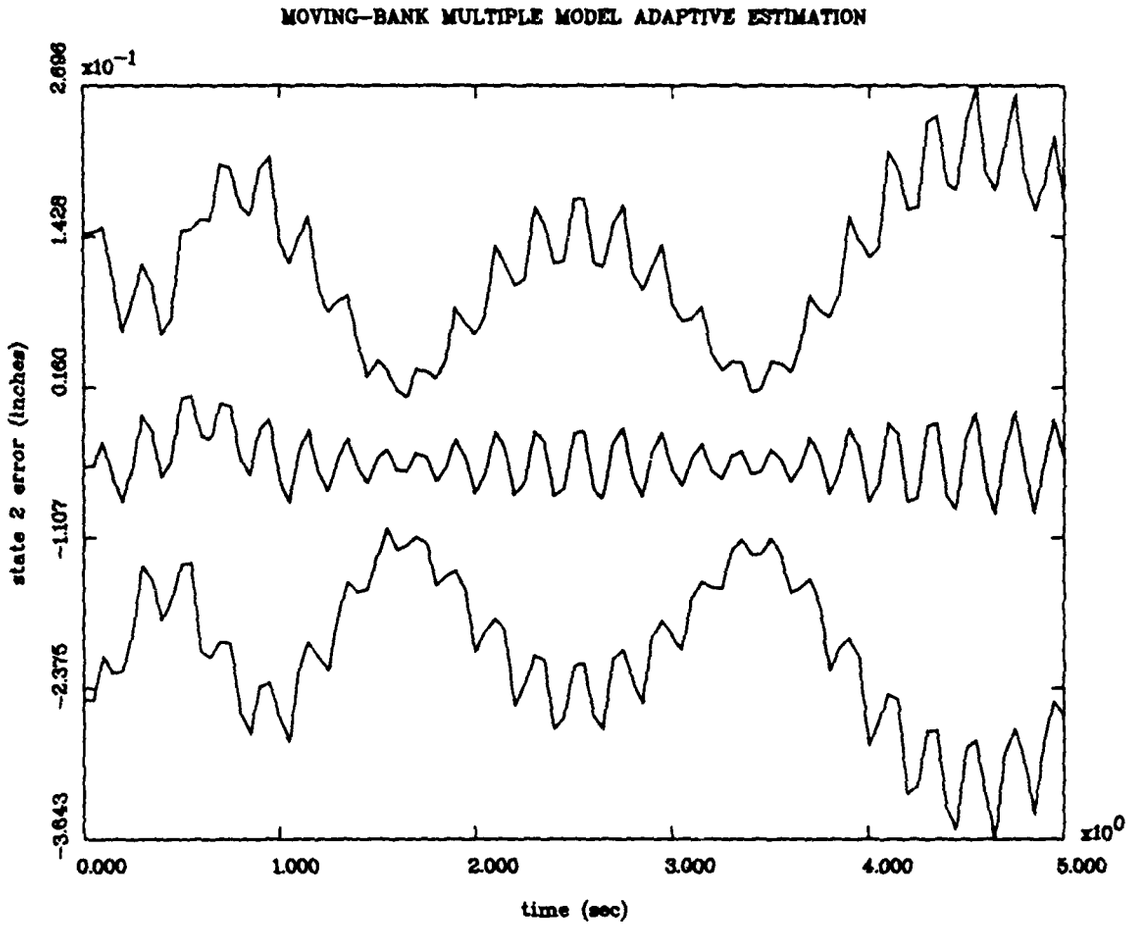

Figure 23(b) State 2
State Estimation Errors
Mean $\pm$ 1σ
Robustness Study
For Single Non-Adaptive Artificially Informed Filter
with 10 Monte Carlo Runs

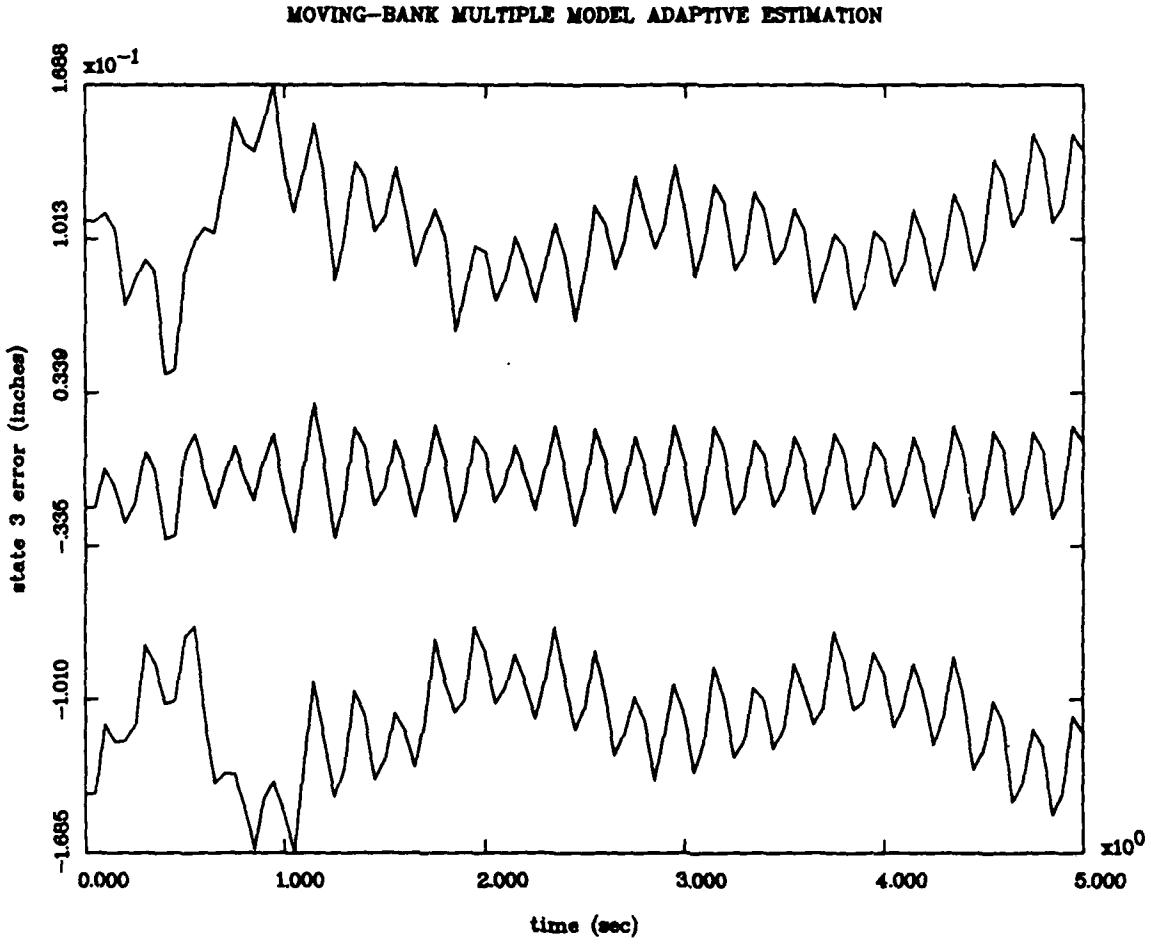

Figure 23(c) State 3
State Estimation Errors
Mean ± 1σ
Robustness Study
For Single Non-Adaptive Artificially Informed Filter
with 10 Monte Carlo Runs

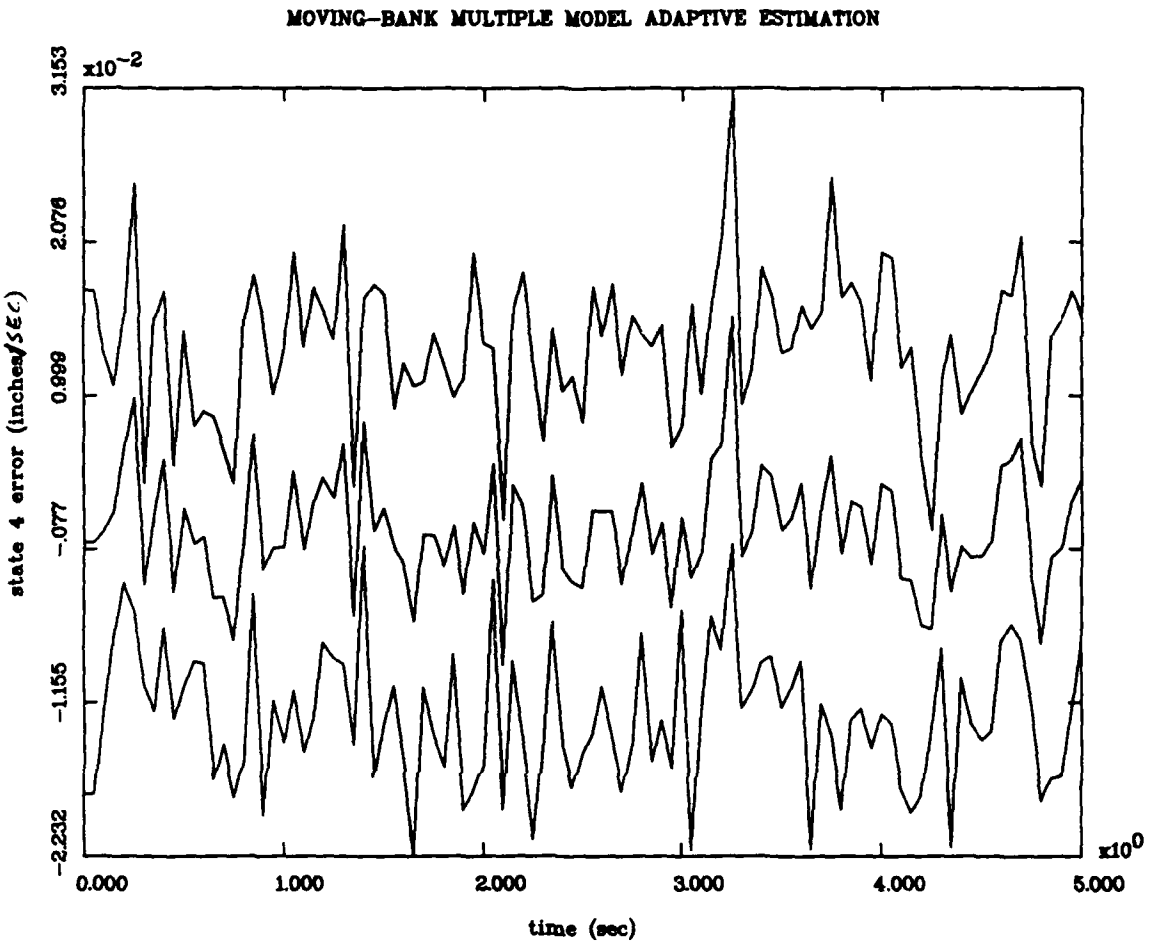

Figure 23(d) State 4
State Estimation Errors
Mean ± 1σ
Robustness Study
For Single Non-Adaptive Artificially Informed Filter
with 10 Monte Carlo Runs

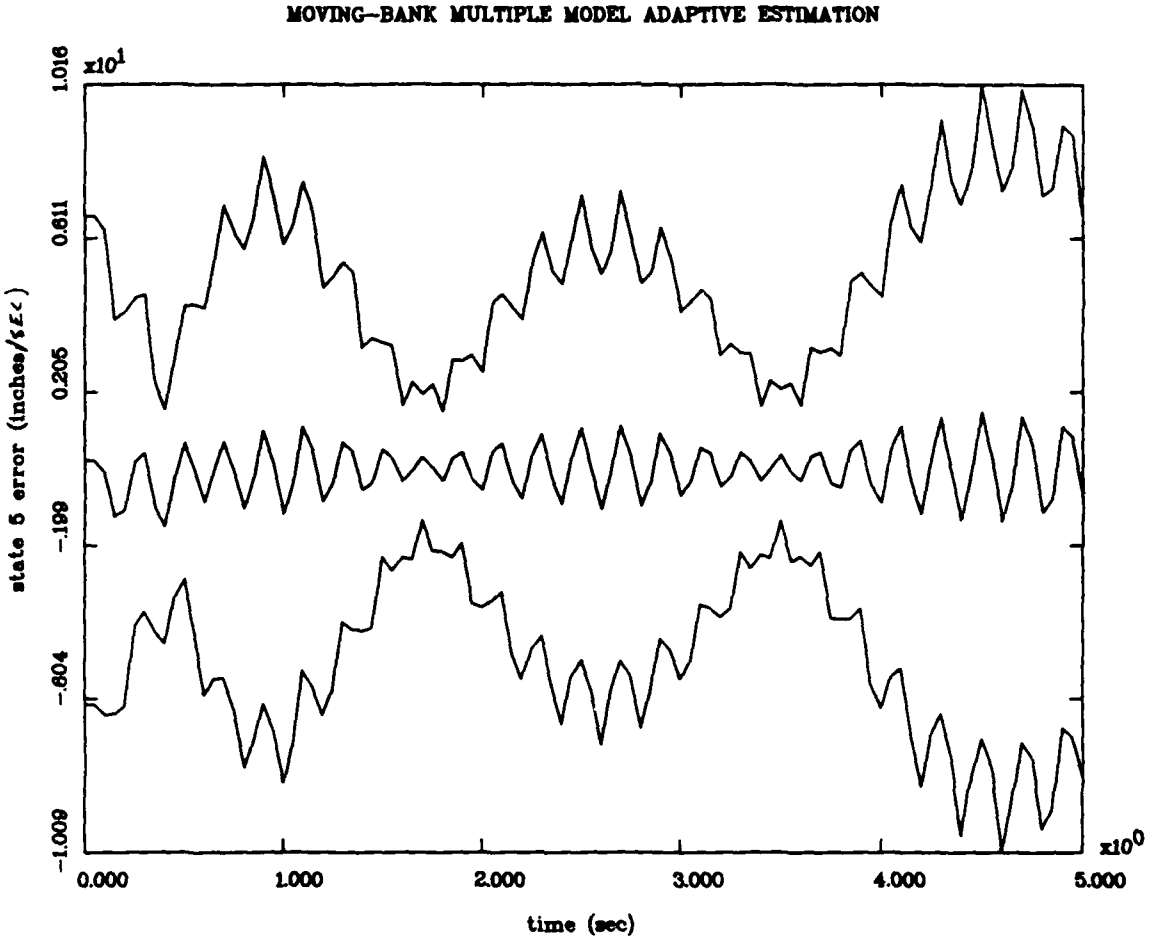

Figure 23(e) State 5
State Estimation Errors
Mean ± 1σ
Robustness Study
For Single Non-Adaptive Artificially Informed Filter
with 10 Monte Carlo Runs

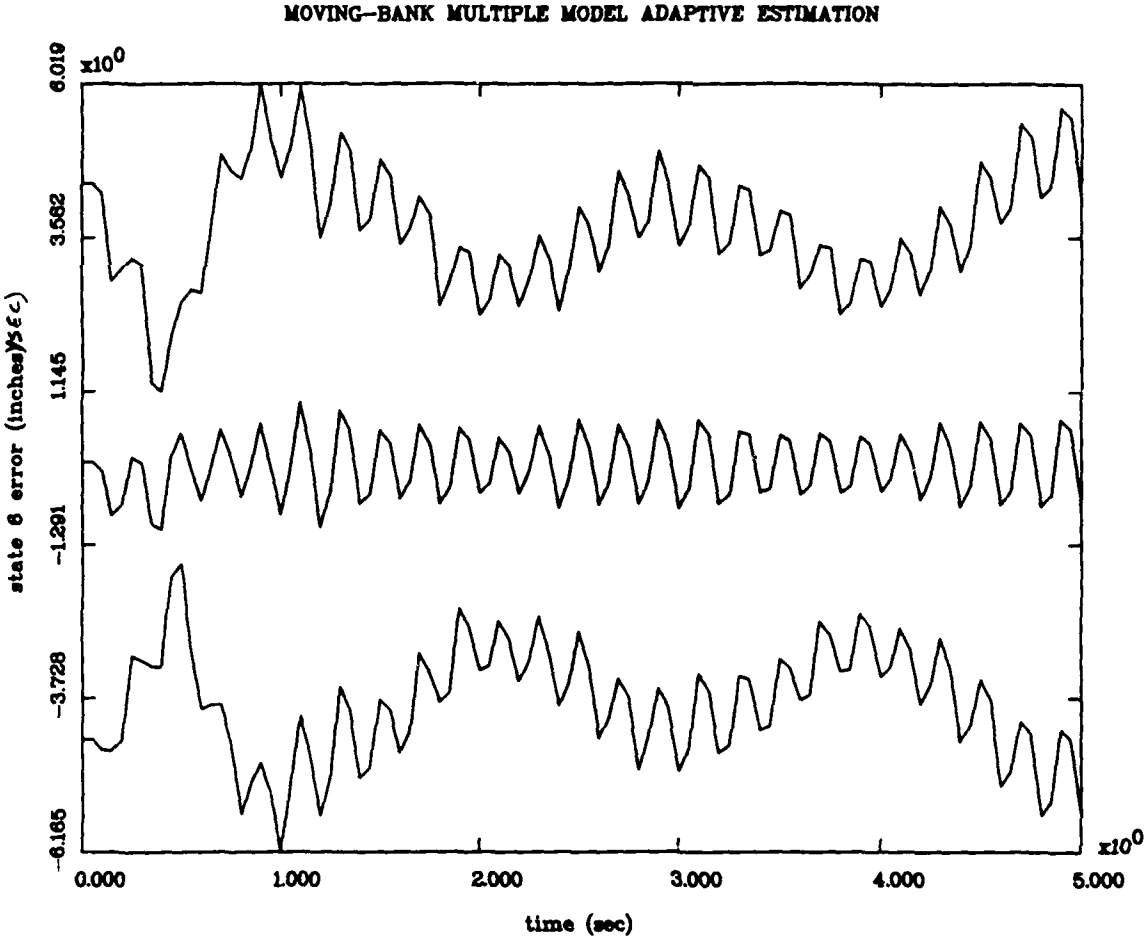

Figure 23(f) State 6
State Estimation Errors
Mean ± 1σ
Robustness Study
For Single Non-Adaptive Artificially Informed Filter
with 10 Monte Carlo Runs

movable bank cannot provide accurate parameter estimates. This inability to provide accurate parameter estimates may well be due to the fact that Q_t increased. Likewise, Lashlee concludes that:

> ...there is a strong correlation between the ability of the moving-bank algorithm to estimate the true parameter and its ability to provide precise state estimates. [9:21]

Doubtlessly, there is a correlation between the ability of the moving bank to provide parameter estimates and its ability to produce state estimates, the question is how strong is this correlation. In light of the results discussed in the noise level determination study, this correlation appears to be weak when the moving bank is initially centered at the correct parameter. Initially centering the moving bank at an incorrect parameter demonstrates a stronger correlation. This demonstration will be discussed in the following paragraphs.

Another area of comparison with Lashlee's thesis is the role the R matrix plays in the algorithms performance. In Lashlee's thesis [8:198] the value of the R matrix played a very large role in the MMAE algorithm's ability to perform. Likewise, in this thesis effort, the values of R_f and R_t play a significant role in the MMAE algorithm's performance. If the R_t matrix is too large, the difference between good and bad filters is masked [9:198]; however, if the R_f matrix is too small, the filter assumes its model for the measured outputs, of the form of Equation (61), is more accurate than it really is. For this thesis, the Q matrix

also needs to be increased to keep the overall Kalman filter gain from decreasing as R is increased.

Centering the moving bank on an incorrect parameter value (5,5) demonstrates the effect of the bank's inability to provide accurate parameter estimates. Figure 24 (note the dither signal is turned off at 0.5 second) shows this effect for the case R1Q1e when the bank is centered at the parameter (5,5) and the true parameter is actually (7,6). The instability of the plots demonstrates the moving bank's inability to provide stable and accurate states estimates. Tuning the R_f matrix and changing the Q matrix did not help at all (a better solution would have been to tune Q_f and leave Q_t as originally set). In fact, for larger R_f and Q values, the numbers generated are so large that the Elxsi cannot handle them and the software generates "floating exception" errors when the Monte Carlo simulation is performed.

Previously, the discussion seemed to conclude that the results of this thesis were in direct opposition to what Lashlee found concerning the correlation between accurate parameter and state estimates; however, the inability of the MMAE algorithm to provide accurate parameter estimates does seem to affect its ability to provide accurate state estimates. Therefore, the results of this thesis <u>do</u> indeed support the results found by Lashlee--that accurate parameter estimates must be generated to provide accurate state estimates.

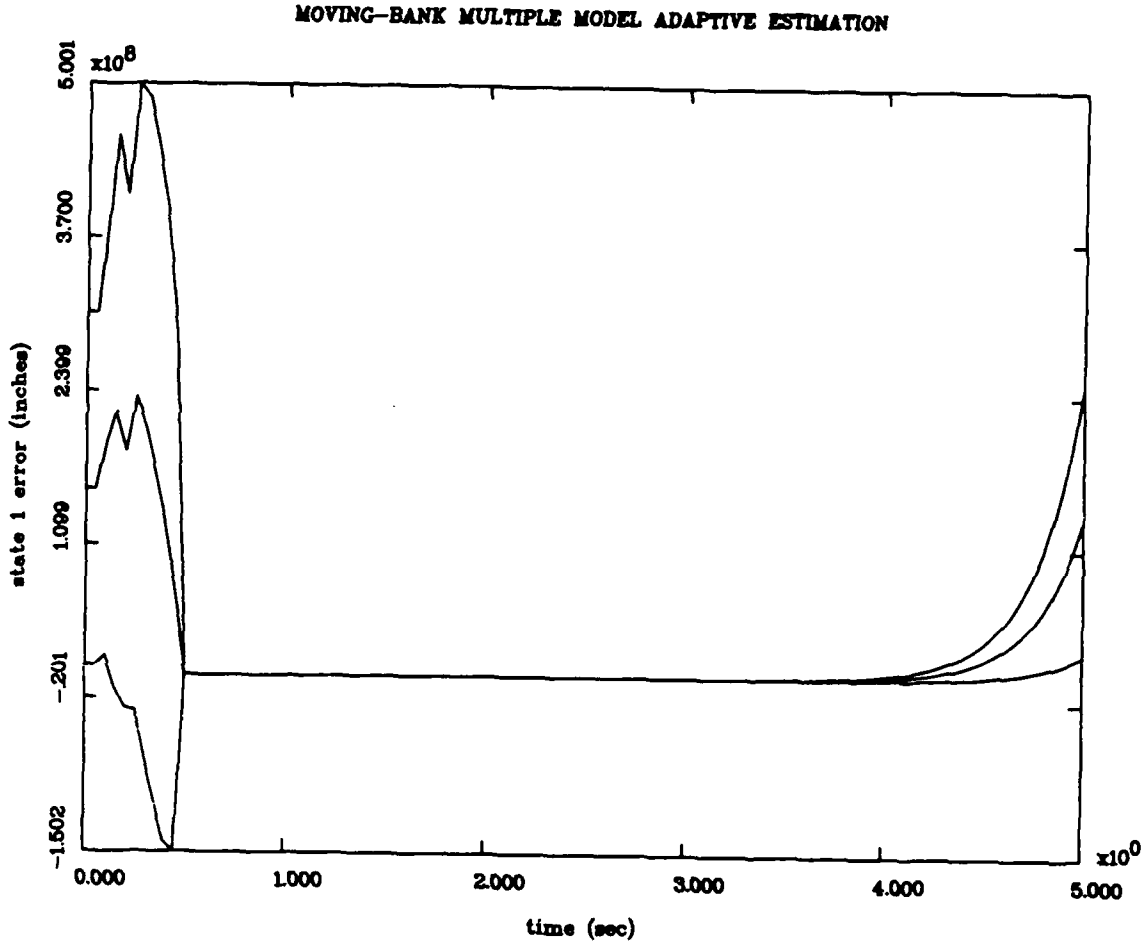

Figure 24(a) State 1
State Estimation Errors
Mean ± 1σ
Robustness Study
R1Q1e Initially Set at Parameter (5,5)
with 10 Monte Carlo Runs

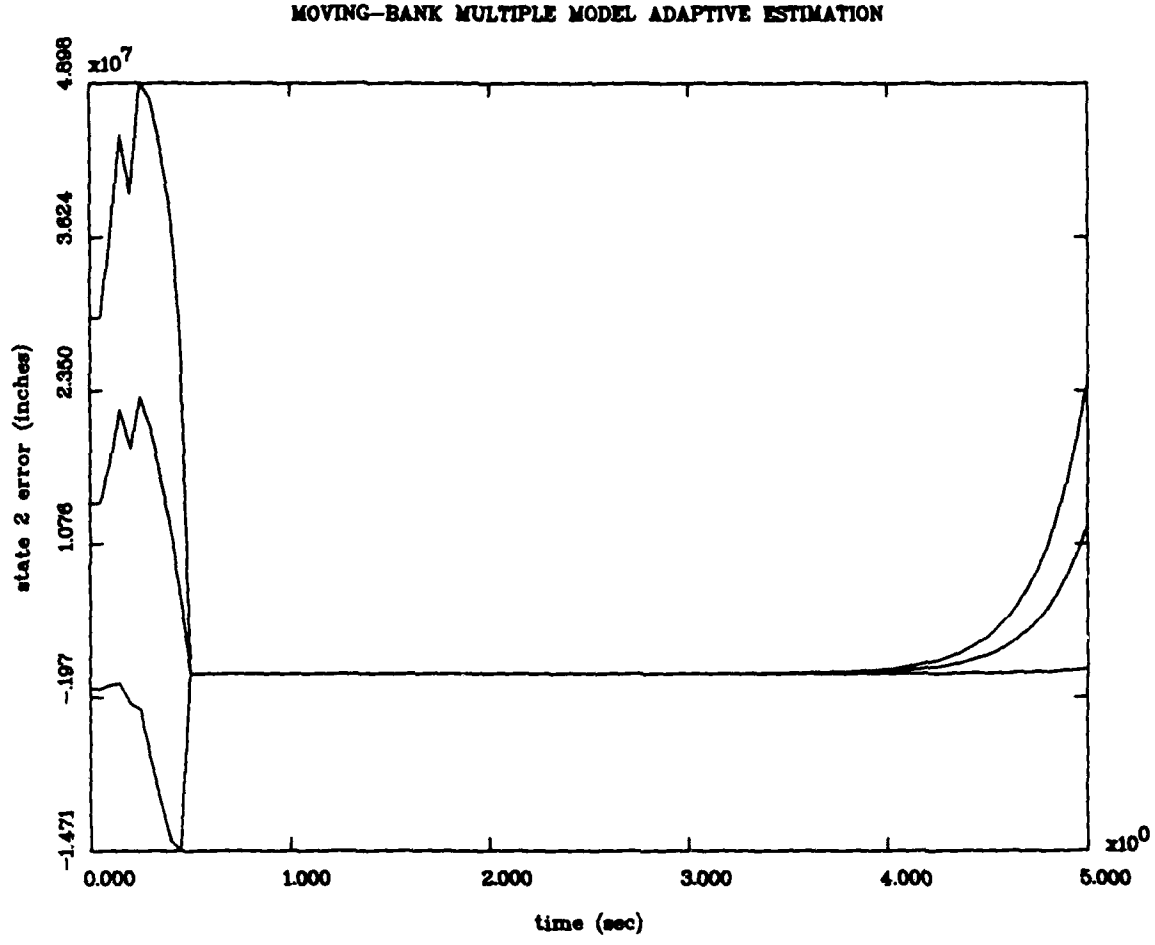

Figure 24(b) State 2
State Estimation Errors
Mean ± 1σ
Robustness Study
R1Q1e Initially Set at Parameter (5,5)
with 10 Monte Carlo Runs

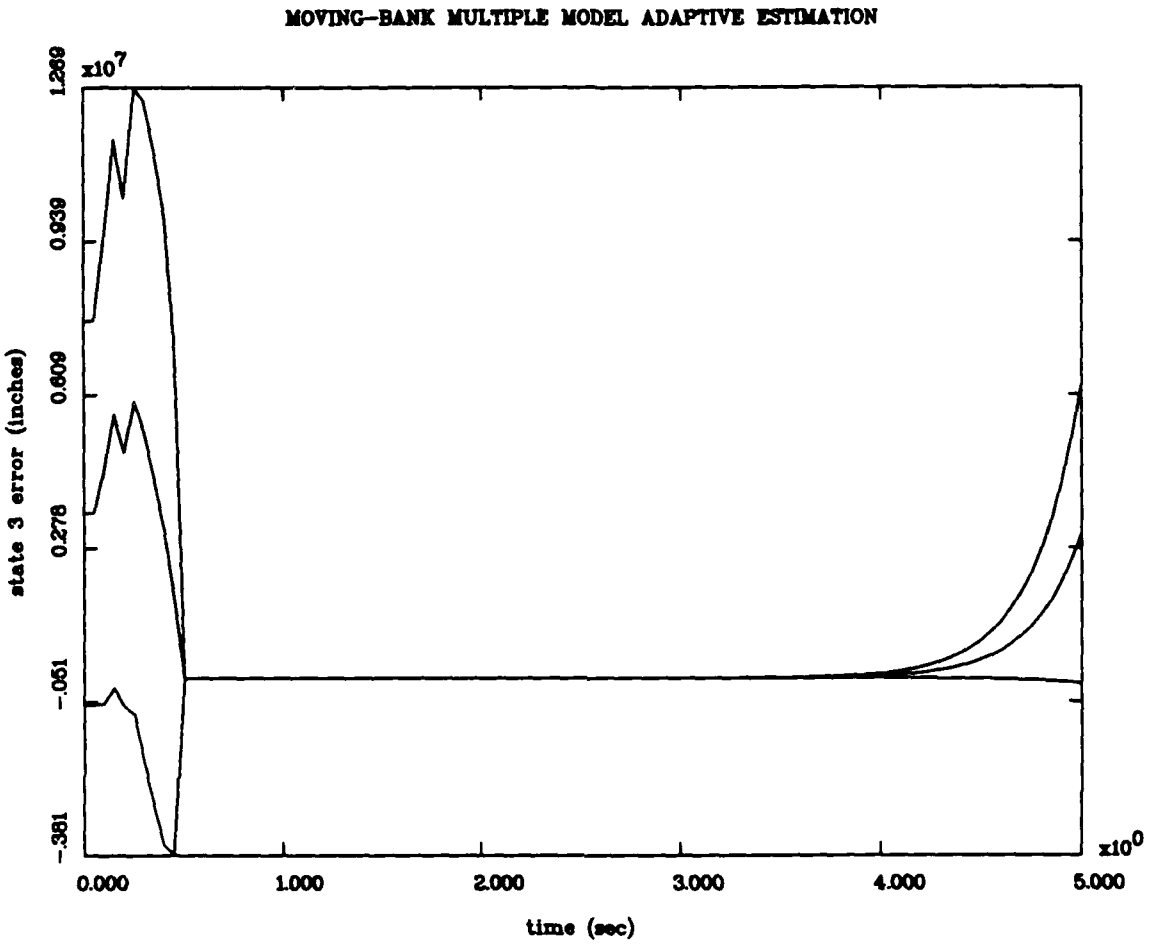

Figure 24(c) State 3
State Estimation Errors
Mean ± 1σ
Robustness Study
R1Q1e Initially Set at Parameter (5,5)
with 10 Monte Carlo Runs

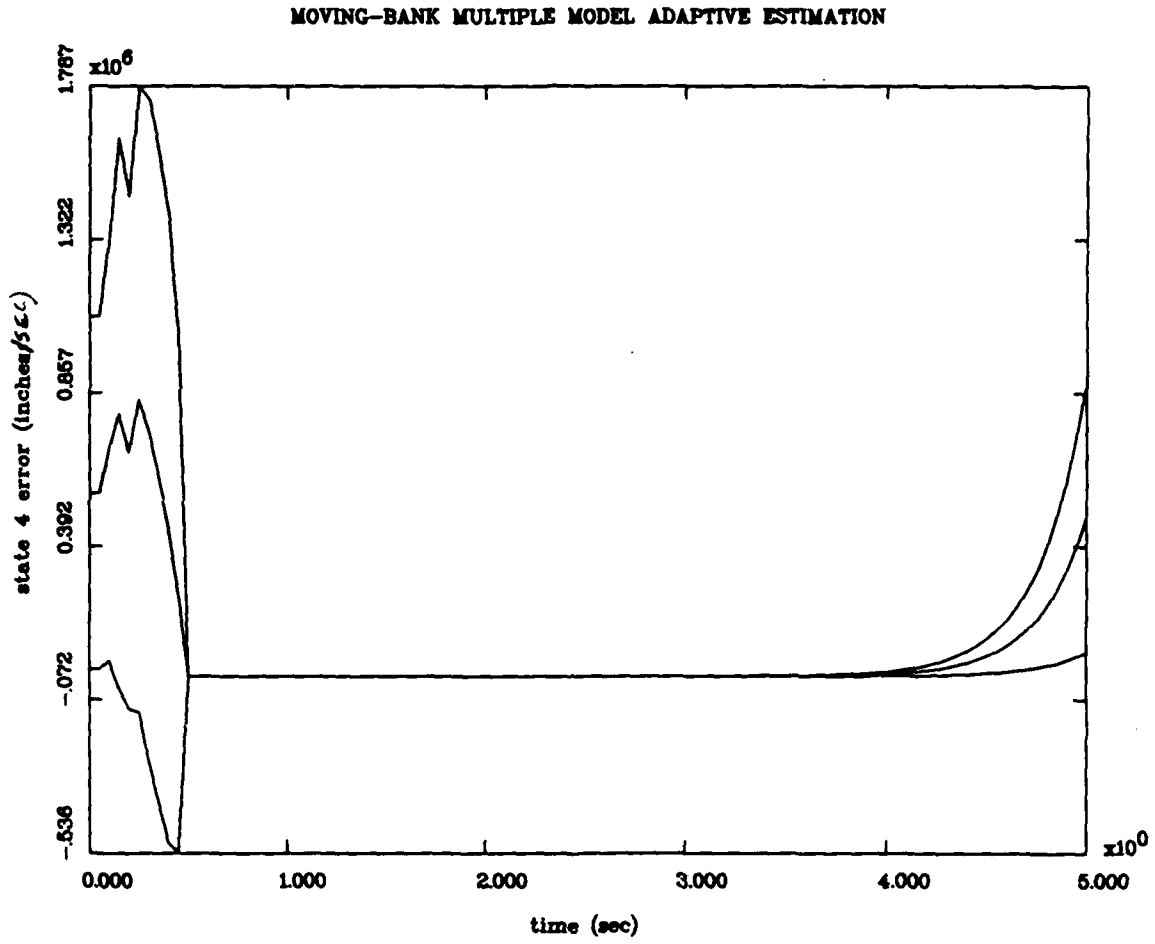

Figure 24(d) State 4
State Estimation Errors
Mean ± 1σ
Robustness Study
R1Q1e Initially Set at Parameter (5,5)
with 10 Monte Carlo Runs

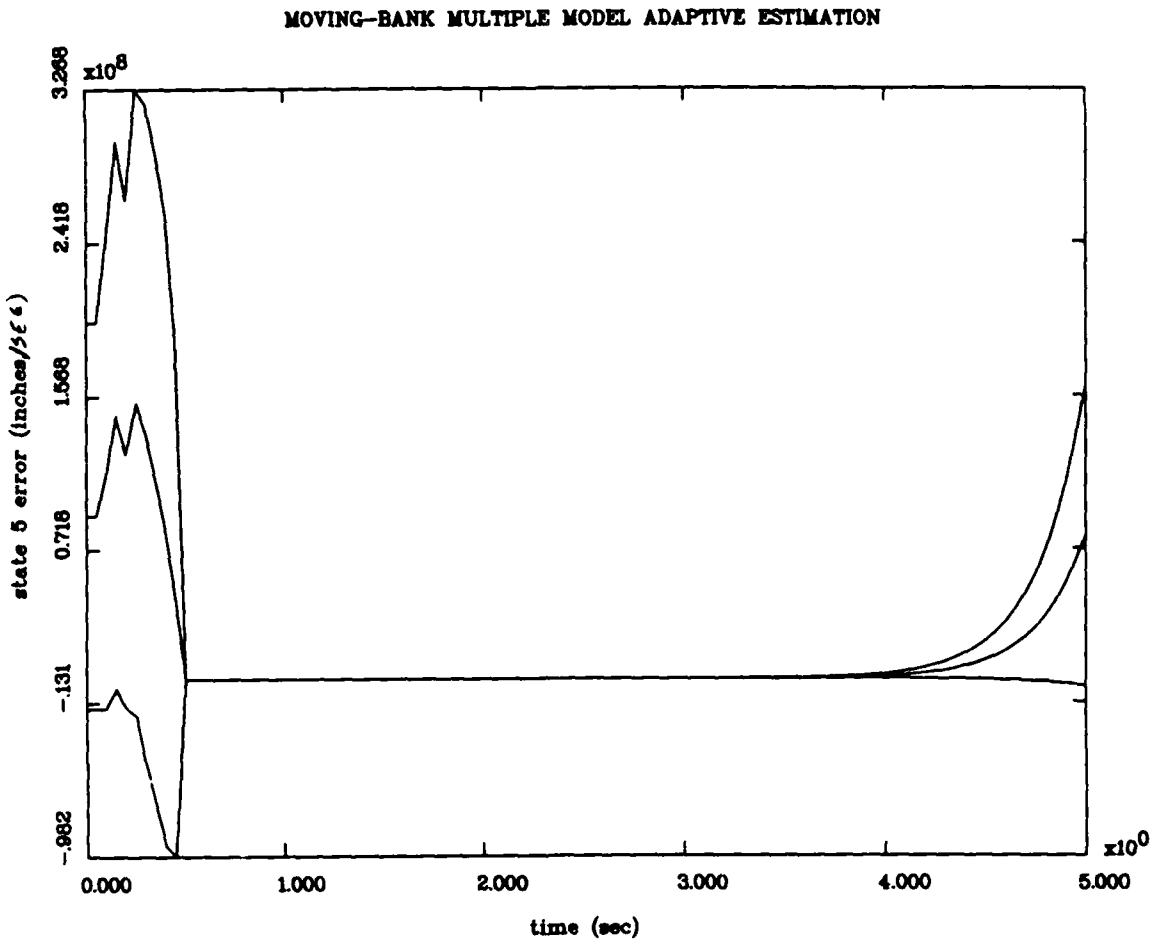

Figure 24(e) State 5
State Estimation Errors
Mean ± 1σ
Robustness Study
R1Q1e Initially Set at Parameter (5,5)
with 10 Monte Carlo Runs

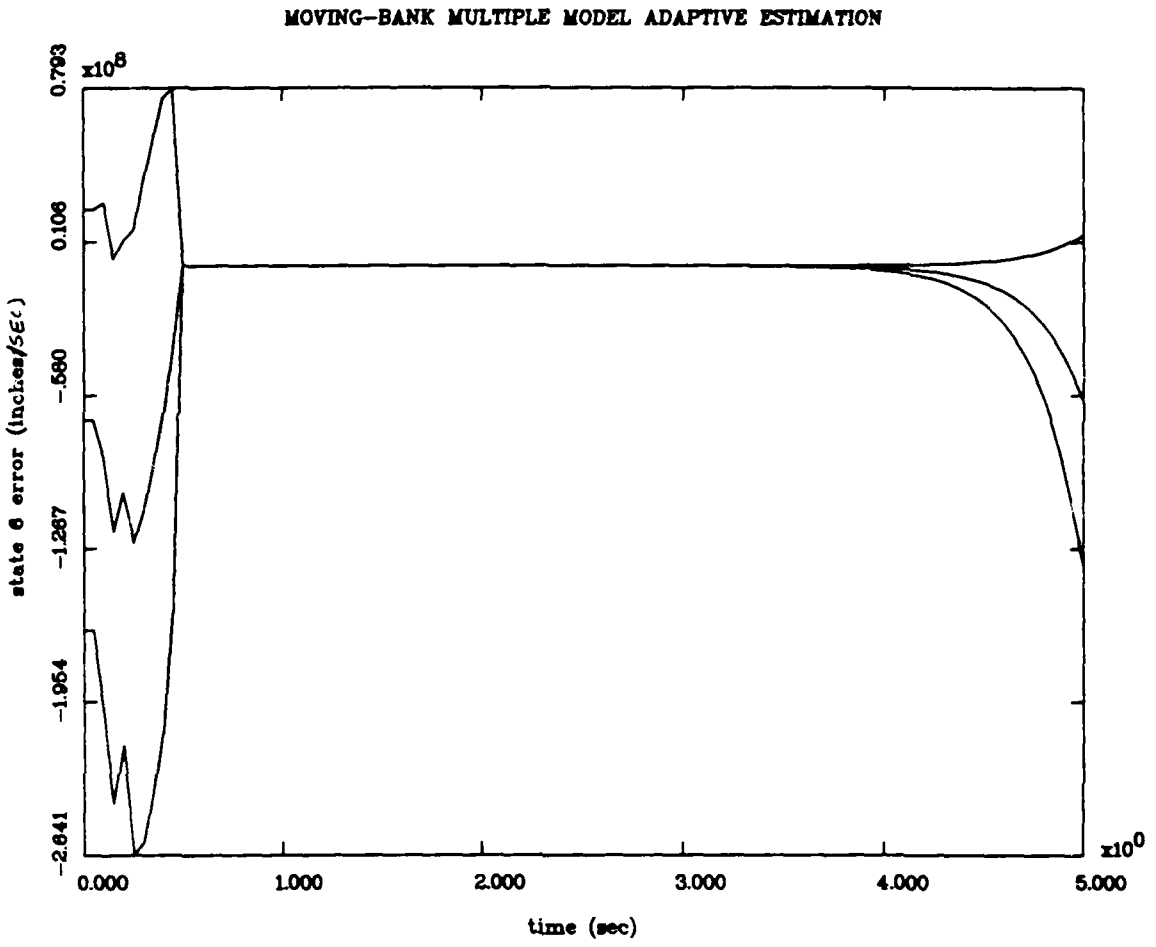

Figure 24(f) State 6
State Estimation Errors
Mean ± 1σ
Robustness Study
R1Q1e Initially Set at Parameter (5,5)
with 10 Monte Carlo Runs

The moving bank of filters was unable to produce stable and accurate state (or parameter) estimates throughout this thesis research when the bank was centered at the false parameter (5,5). This result demonstrates the totally confounding effect the unmodeled states have on the ability of the moving bank algorithm to provide adaptive estimation for the chosen R_f, R_t, Q_t, and Q_f. A close examination of the 24 states involved in the truth model shows that the velocities corresponding to modes 4 through 9 are of the same order of magnitude as the positions and velocities corresponding to the rigid body mode and the first two bending modes (modes 2 and 3). This "equivalence" among the magnitudes of the modes means that some of the lower bending modes are <u>not</u> negligible as they were originally assumed. These unmodeled modes (states) seem to have a direct impact on the ability of the MMAE algorithm to provide stable and accurate state and parameter estimates.

The results found during the noise level determination study indicate that, when the bank is initially centered on the true parameter point, stable and accurate state estimates can be produced. However, this section demonstrates that, when the bank is initially centered at a false parameter point, no stable or accurate state estimates can be produced.

A basic assumption is made that the MMAE algorithm does not <u>know</u> the <u>true</u> parameter when initialized in the real world. This

assumption means that the moving bank should be able to provide accurate and stable estimates when the bank is initially centered at a parameter value that is not the true parameter, at least for the chosen values of R_t, R_f, Q_t, and Q_f. This thesis research has demonstrated that the MMAE algorithm cannot provide the required estimates when initially centered at a false parameter. Therefore, the unmodeled states do indeed seriously degrade the adaptation process, since adequate performance has been previously demonstrated (by Lashlee) when using a 6-state truth model.

5.2.4 Controller Study

The controller study is performed using the MMAC algorithm shown in Figure 2. The object of this study is to determine whether or not adequate MMAC state regulation can be performed in the presence of the unmodeled states. Unlike the previous sections, the matrix Q_f does vary from Q_t for this study. This true Q_f and R_f tuning is implemented because the results of the previous sections indicate a need for Q_f tuning as well as R_f tuning to produce stable parameter and state estimation. As previously mentioned, this true filter tuning was not implemented in the previous studies because it was assumed that mainly the R_f matrix affected the performance of the adaptive algorithm. Originally, R_f is set to $R_{f\text{-}old}$, which is the value found in Section 4.4.2, and Q_f is set to Q_t, which is the value found by Lashlee [8:94]. The matrix Q_t is not changed because it was

found to provide stable and accurate parameter and state estimates for Lashlee's research. The notation used in this section for each case of Q_f and R_f is similar to the notation discussed previously except that "c" is used to indicate control.

As stated previously, the Monte Carlo analysis using the MMAC algorithm provided no stable control for any of the cases which provided stable estimation for the noise level determination study. In fact, stable control was <u>not</u> provided for any of the Q_f and R_f values used. The reason for this is unknown. This is another area of investigation for further research. Time limitations for this thesis research precluded further study of the MMAC controller.

Note that the dither signal is also "turned off" at 0.5 second for the control study. Likewise, the controller is not "turned on" until 0.5 second, so the algorithm is basically functioning "open loop" while the dither signal is activated.

As with Lashlee's thesis [8:141-143], the state weighting and control weighting matrices must be tuned for each parameter in the parameter space. Lashlee [8:200] recommended that the state and control weighting matrices be tuned for each parameter point. This tuning process entails a "tuning study" for each of the 100 parameter values where the tuning is performed on the state weighting matrix "...until the rms error on the true states

stop changing drastically" [8:142-143]. This process is extremely time consuming and was not done for this thesis research. For this research, no state and control weighting matrix tuning was performed. The state and control weighting matrices were simply kept at the tuning values for the parameter (7,6) that were found by Lashlee [8:143].

The result of some Q_f and R_f tuning of the MMAC is presented by the plots of Figure 25. This figure represents only one of the cases researched. These plots show the results for the case where $R_f = 0.5 * R_{f\text{-}old}$ and $Q_f = 0.5 * Q_t$ (denoted by R0.5Q0.5c for this case). The first plot of this figure appears stable, but the other plots appear to have an increasing pseudo-sinusoidal standard deviation. Further investigation, done by increasing the run time from 5 seconds to 10 and 20 seconds, does indeed show an unstable response. These plots are represented by Figure 26 and 27. Larger values of Q_f and R_f produced unstable control as did smaller values. Only this one case is presented in this thesis because it quite readily sums up the results found during this thesis research.

Note that a much better representation of the MMAC performance would have been to show the states (and the regulation of those states to zero) in Figure 25 and Figure 26. Unfortunately, the sample statistics of the states were not

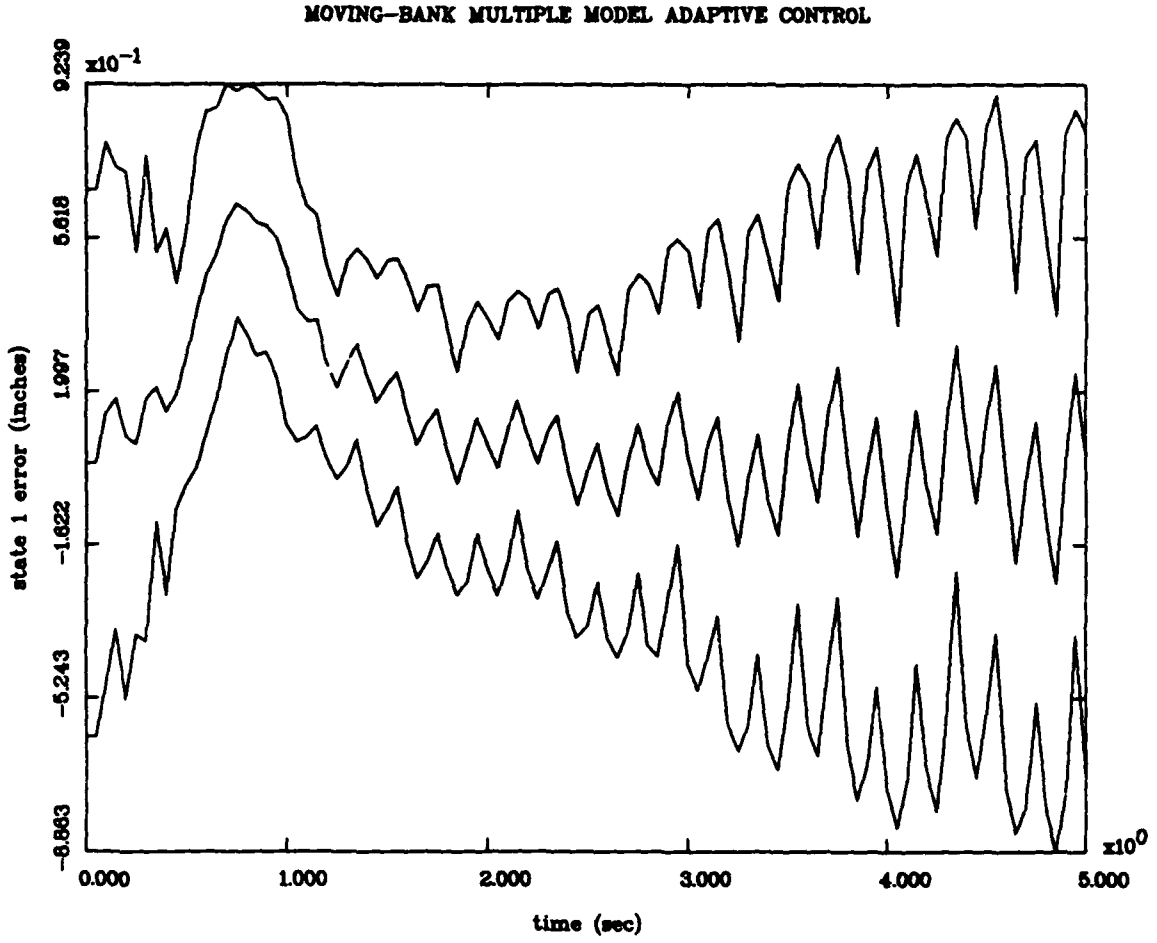

Figure 25(a) State 1
State Estimation Errors
Mean ± 1σ
Robustness Study
R0.5Q0.5c Case
with 10 Monte Carlo Runs

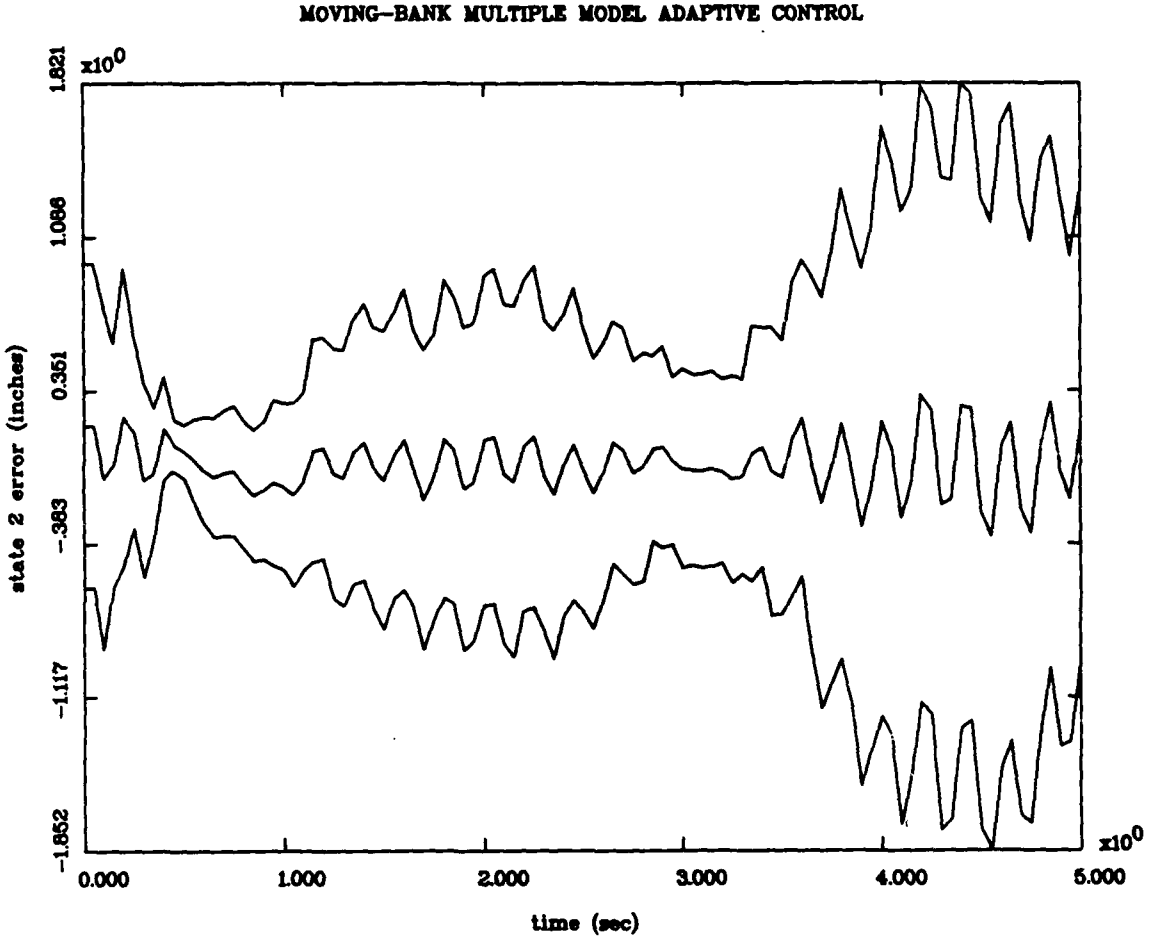

Figure 25(b) State 2
State Estimation Errors
Mean ± 1σ
Robustness Study
R0.5Q0.5c Case
with 10 Monte Carlo Runs

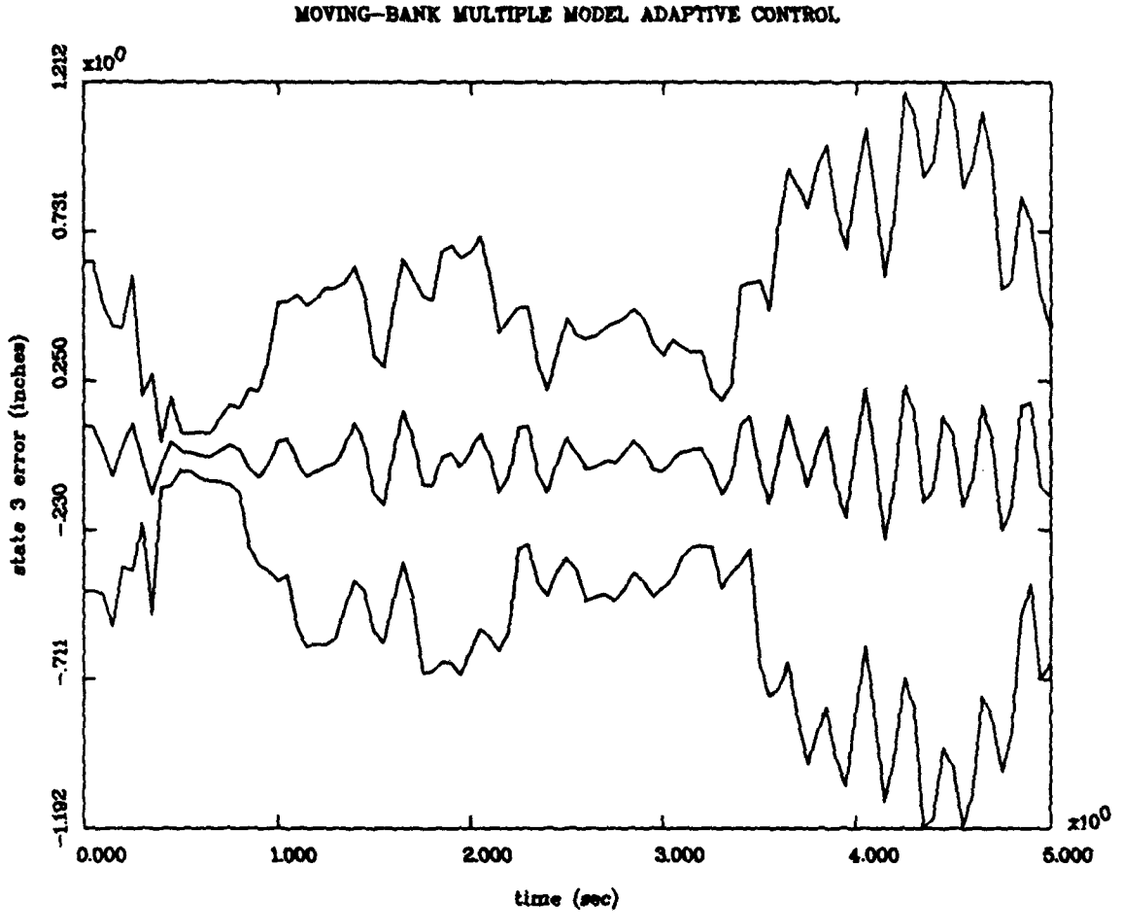

Figure 25(c) State 3
State Estimation Errors
Mean ± 1σ
Robustness Study
R0.5Q0.5c Case
with 10 Monte Carlo Runs

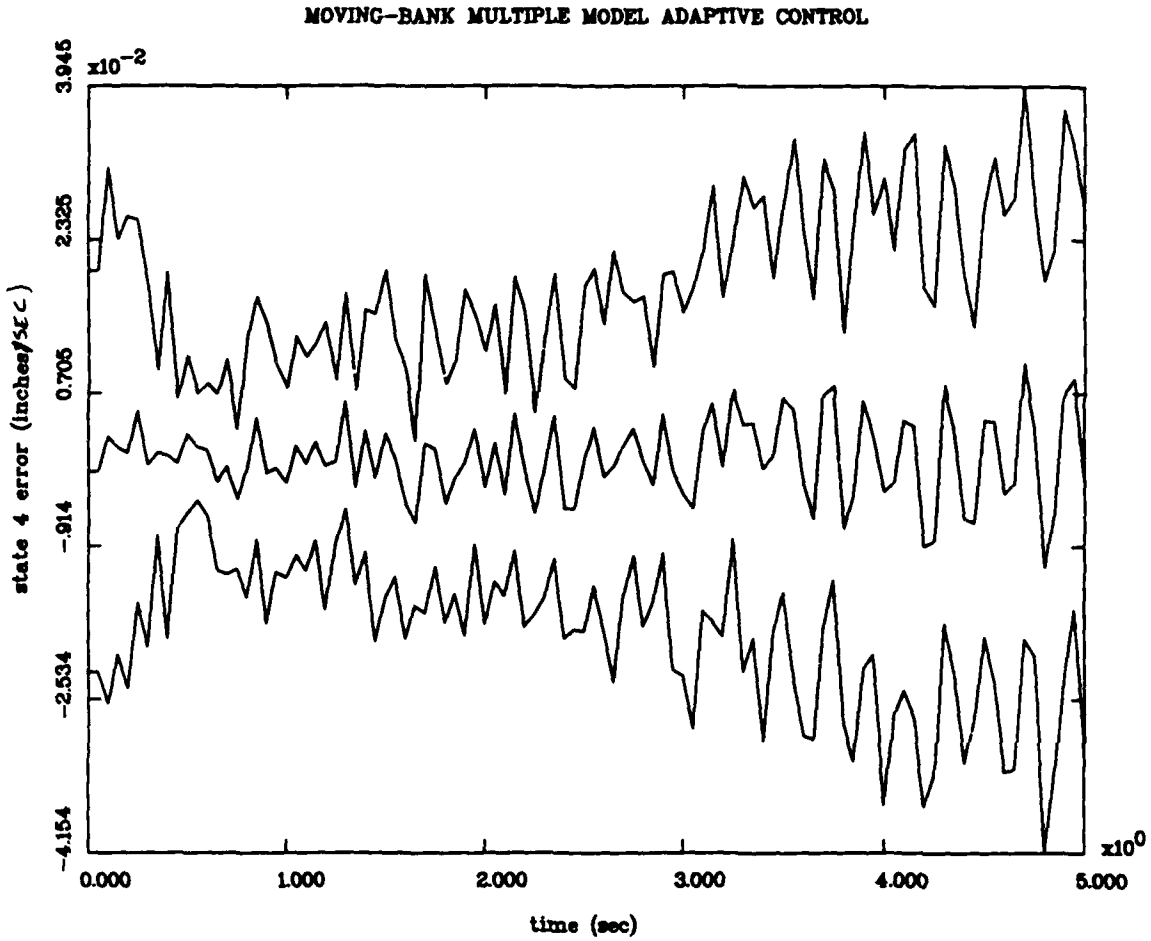

Figure 25(d) State 4
State Estimation Errors
Mean ± 1σ
Robustness Study
R0.5Q0.5c Case
with 10 Monte Carlo Runs

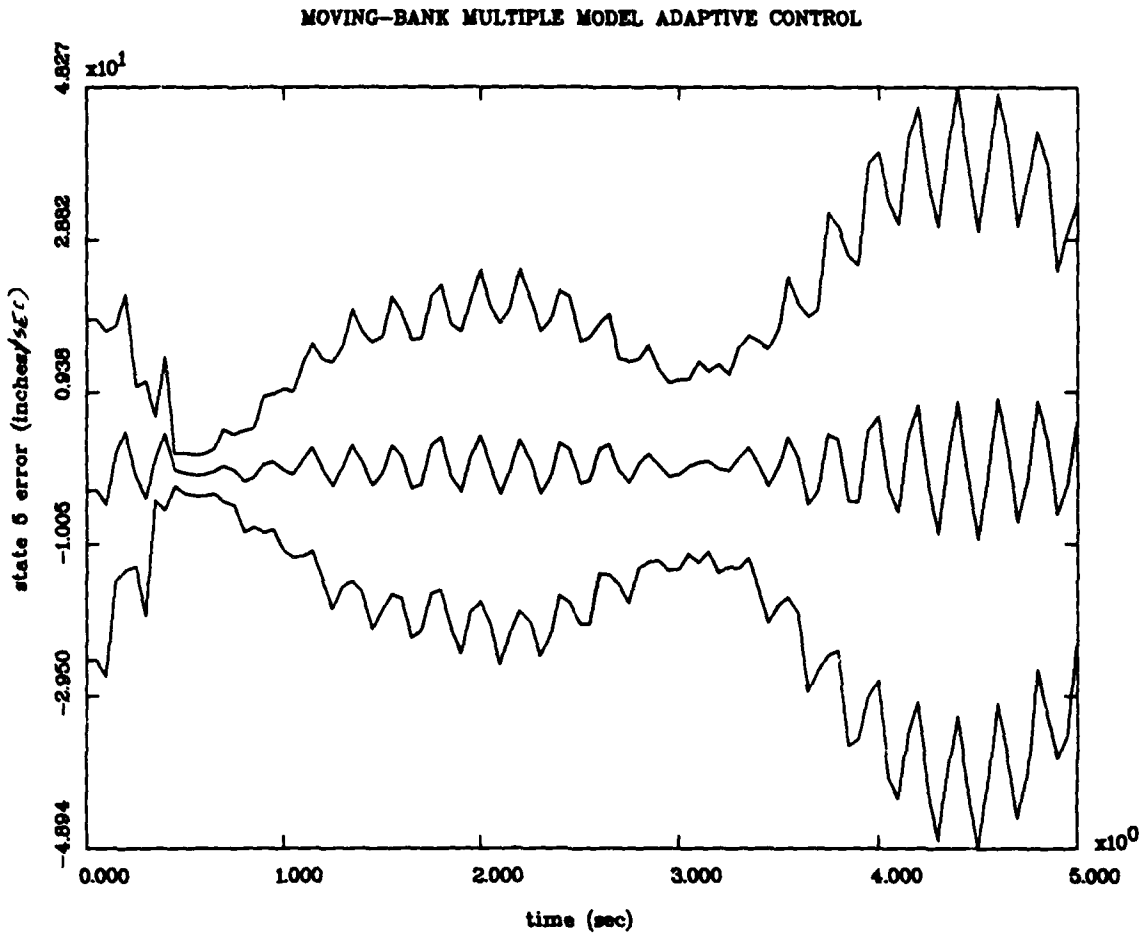

Figure 25(e) State 5
State Estimation Errors
Mean ± 1σ
Robustness Study
R0.5Q0.5c Case
with 10 Monte Carlo Runs

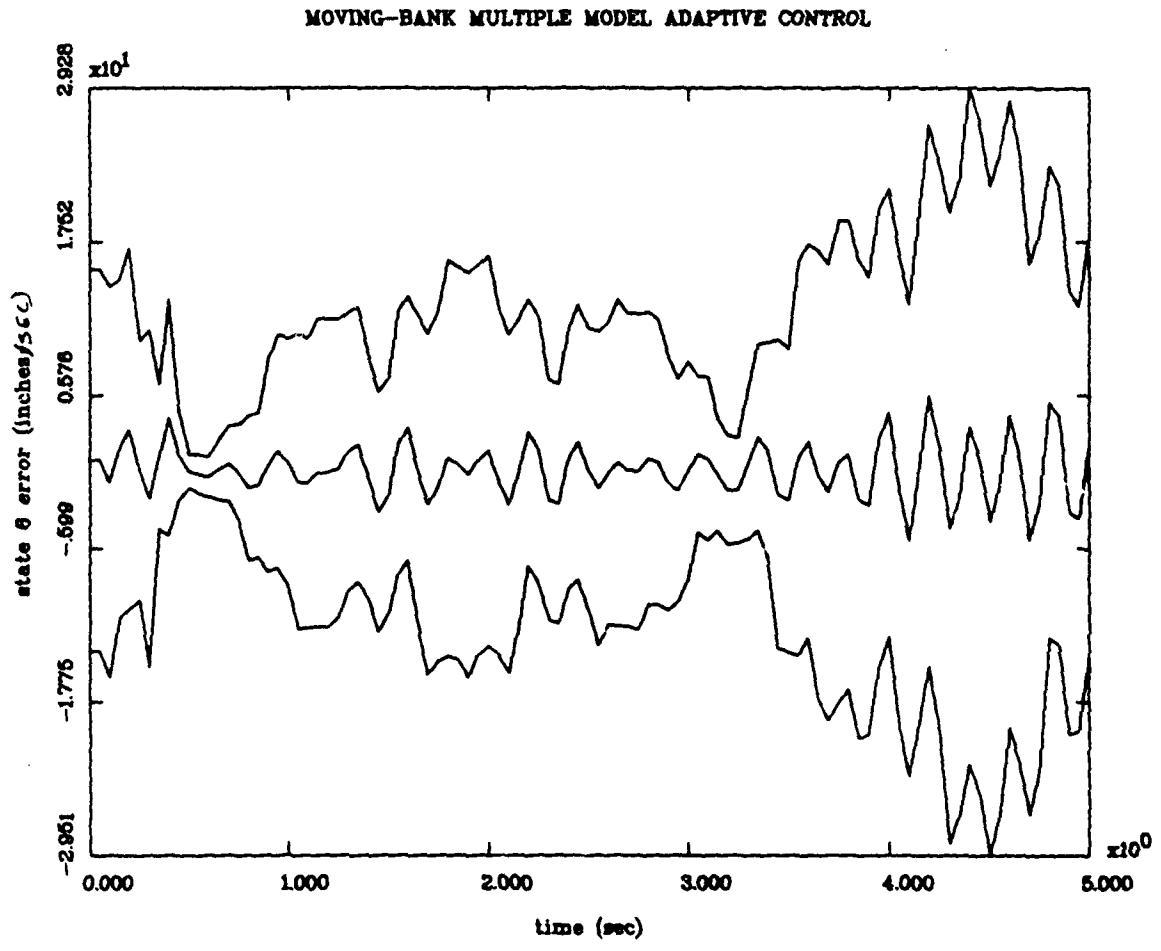

Figure 25(f) State 6
State Estimation Errors
Mean ± 1σ
Robustness Study
R0.5Q0.5c Case
with 10 Monte Carlo Runs

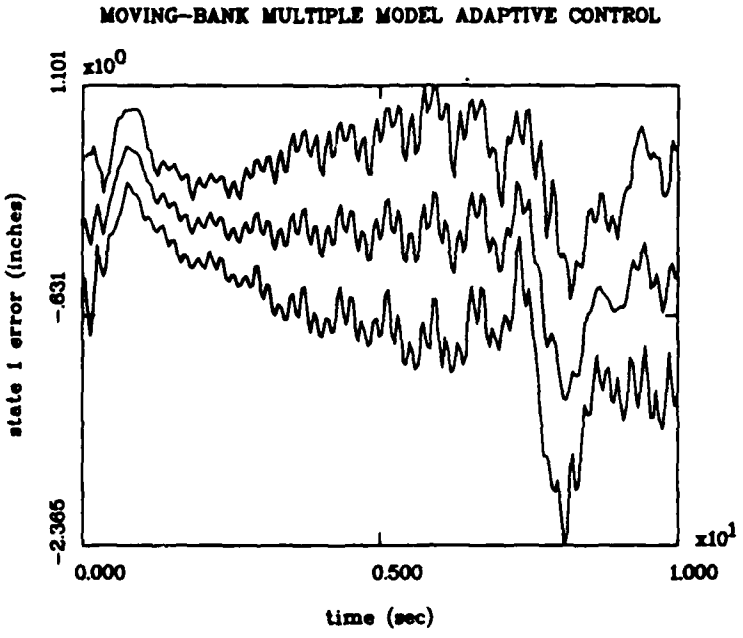

Figure 26(a) State 1
State Estimation Errors
Mean ± 1σ
Robustness Study
R0.5Q0.5c Case
10 Second Run

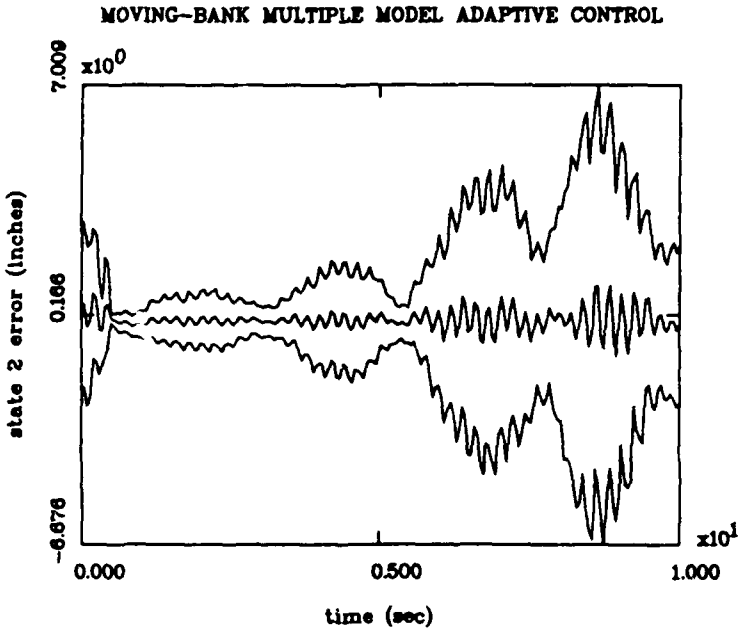

Figure 26(b) State 2
State Estimation Errors
Mean ± 1σ
Robustness Study
R0.5Q0.5c Case
10 Second Run

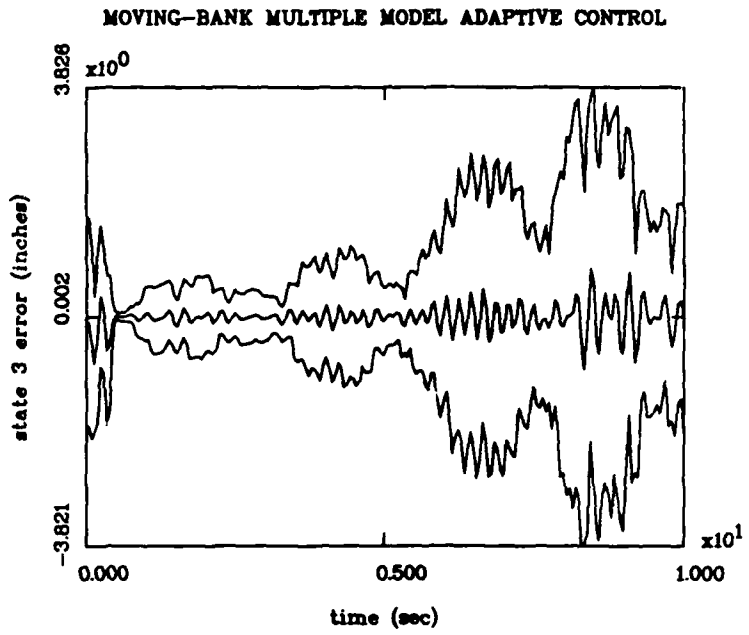

Figure 26(c) State 3
State Estimation Errors
Mean ± 1σ
Robustness Study
R0.5Q0.5c Case
10 Second Run

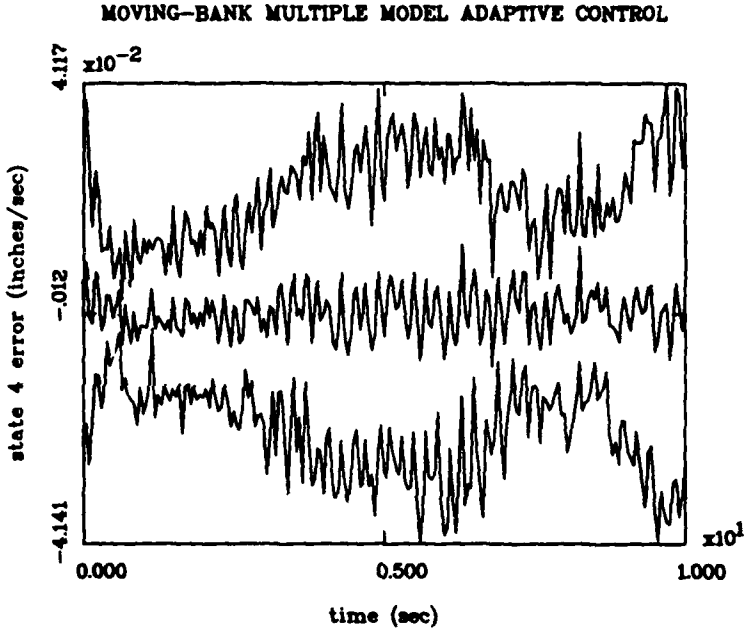

Figure 26(d) State 4
State Estimation Errors
Mean ± 1σ
Robustness Study
R0.5Q0.5c Case
10 Second Run

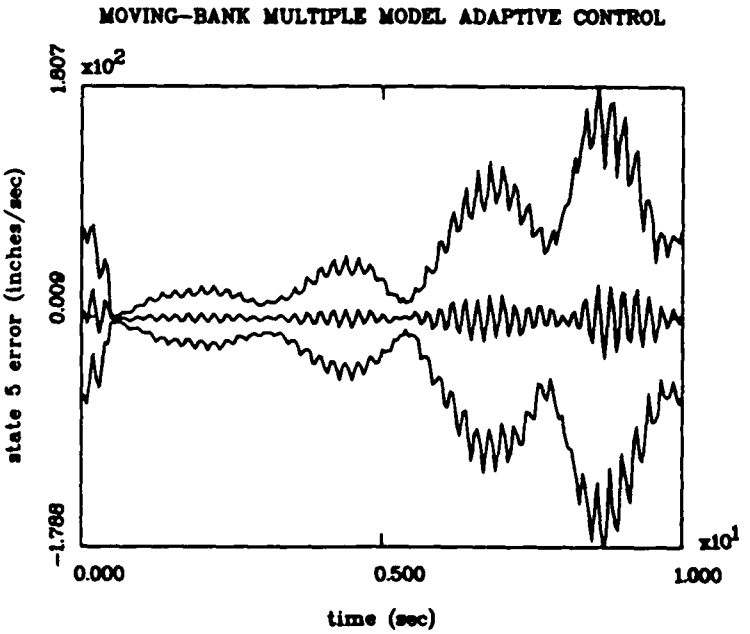

Figure 26(e) State 5
State Estimation Errors
Mean ± 1σ
Robustness Study
R0.5Q0.5c Case
10 Second Run

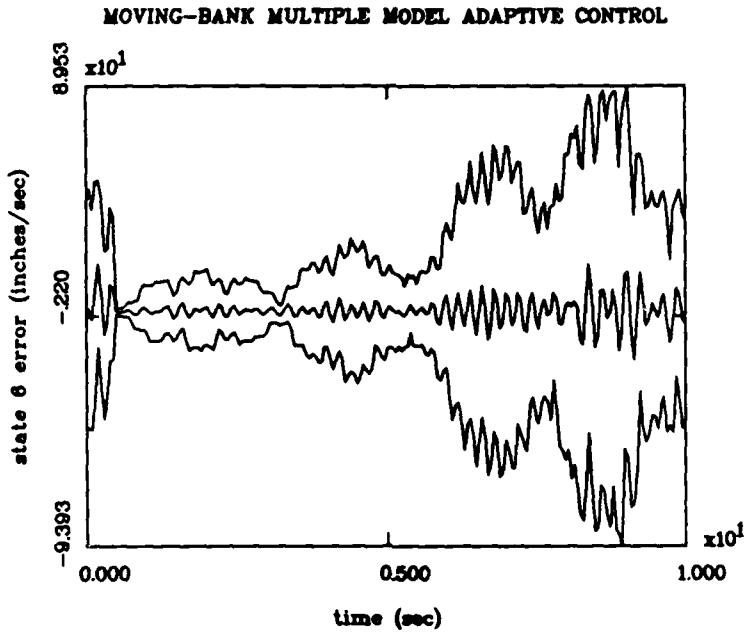

Figure 26(f) State 6
State Estimation Errors
Mean ± 1σ
Robustness Study
R0.5Q0.5c Case
10 Second Run

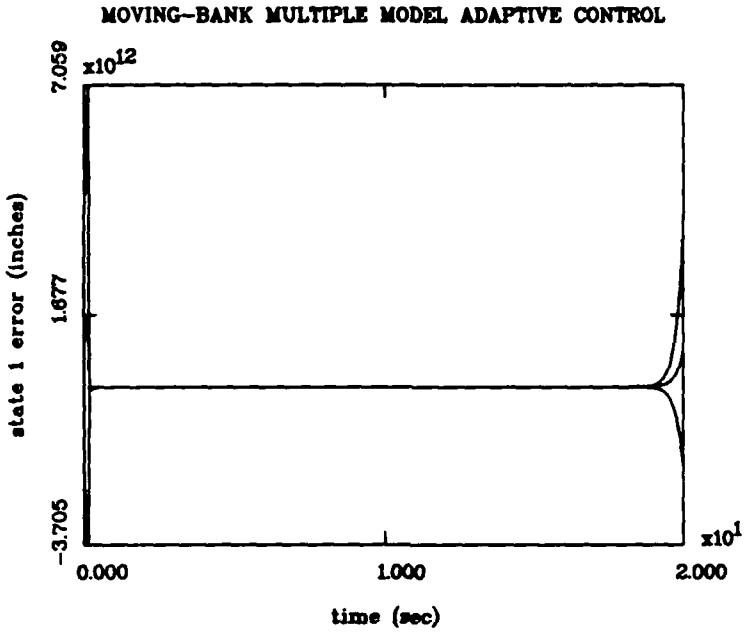

Figure 27(a) State 1
State Estimation Errors
Mean ± 1σ
Robustness Study
R0.5Q0.5c Case
20 Second Run

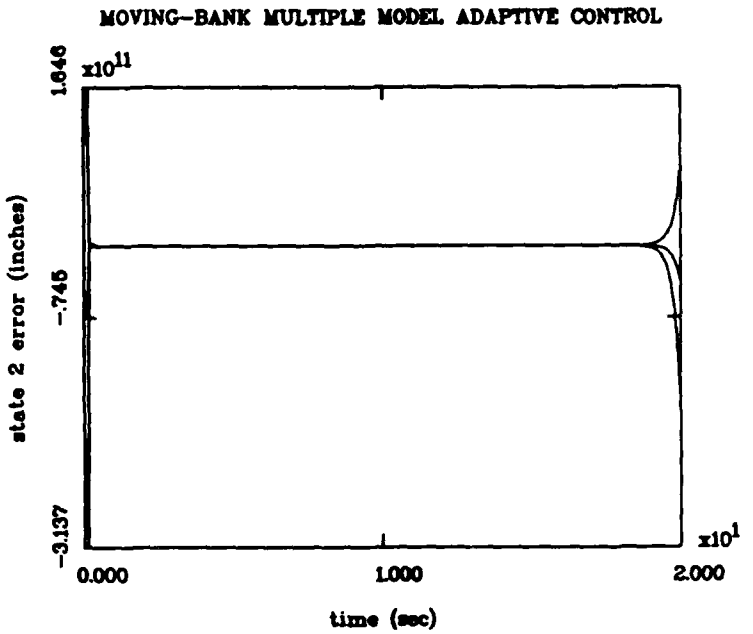

Figure 27(b) State 2
State Estimation Errors
Mean ± 1σ
Robustness Study
R0.5Q0.5c Case
20 Second Run

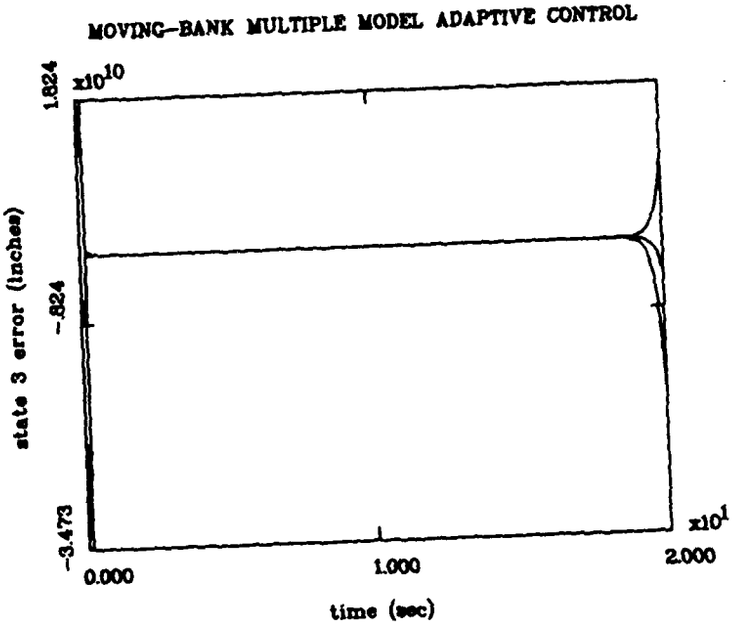

Figure 27(c) State 3
State Estimation Errors
Mean ± 1σ
Robustness Study
R0.5Q0.5c Case
20 Second Run

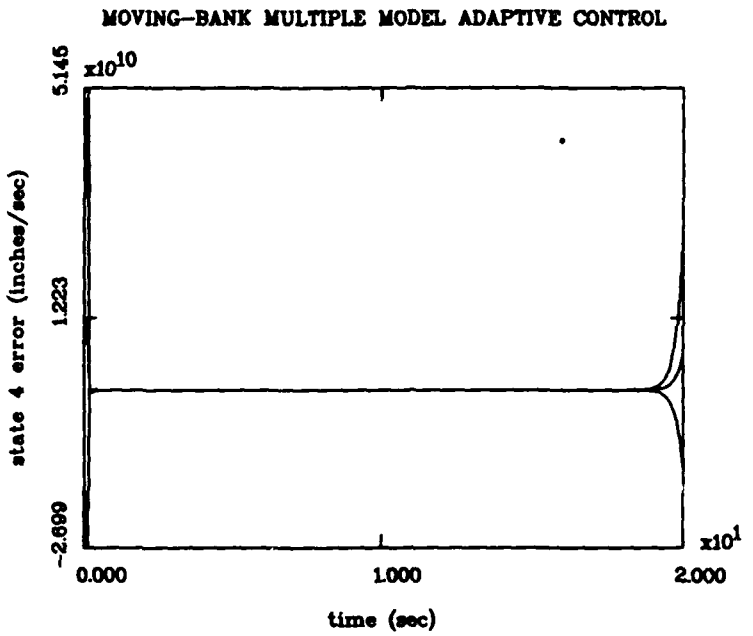

Figure 27(d) State 4
State Estimation Errors
Mean ± 1σ
Robustness Study
R0.5Q0.5c Case
20 Second Run

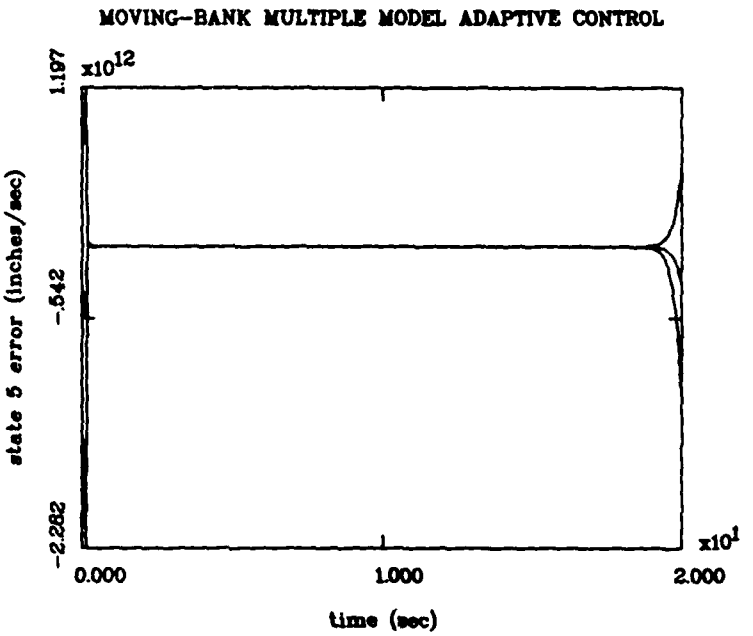

Figure 27(e) State 5
State Estimation Errors
Mean ± 1σ
Robustness Study
R0.5Q0.5c Case
20 Second Run

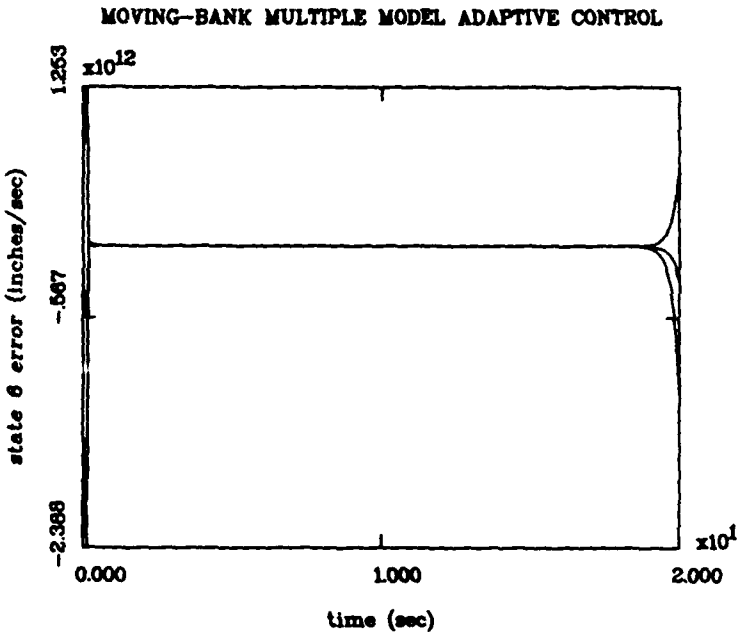

Figure 27(f) State 6
State Estimation Errors
Mean ± 1σ
Robustness Study
R0.5Q0.5c Case
20 Second Run

extracted and the plots not generated. The state estimation error is therefore used to represent a measure of control. Note that for both Figures 26 and 27, the simulation still only employs 10 Monte Carlo runs.

The objective of this study is achieved by demonstrating that no stable control can be provided in the presence of the unmodeled states for the Q_t, Q_f, R_t and R_f chosen. The controller study is definitely an area that requires more study.

5.3 Summary

The objective of the thesis is achieved in this chapter by demonstrating that the effects of the unmodeled states do seriously degrade the ability of the MMAE process and totally confounds the MMAC process. Apparently, the influence of the unmodeled states corresponding to modes 4 through 9 have a serious effect on the moving bank's ability to provide adaptive state and parameter estimates for the chosen Q_f, R_f, Q_t, and R_t.

Only when the moving bank is initially centered at the true parameter does the algorithm generate stable MMAE results. Likewise, proper tuning of the R_f and Q_f matrices is required for accurate performance. A better way to perform the noise level determination study and robustness analysis would have been to allow Q_f to change as well as R_f. This is recommended for further research. Likewise the availability of parameter

estimate plots and state regulation plots would have shed more light on the moving-bank algorithm's performance.

VI. Conclusion and Recommendations

6.1 Introduction

The robustness analysis performed in this thesis of the MMAE and MMAC algorithms demonstrates that the unmodeled states cause a serious degradation of the algorithms' abilities to provide adaptive estimation and control. This thesis research provides a strong base for follow-on thesis efforts.

6.2 Conclusions

The magnitudes of the velocity states associated with modes 4 through 9 (6 of the unmodeled modes) are as large as the magnitudes of the position and velocity states associated with the modeled modes (rigid body and first two bending modes). These unmodeled modes are incorrectly assumed "negligible" and cause a serious degradation of the ability of the MMAE process and a total confounding of the MMAC process. In effect, the algorithms demonstrate very little robustness in the presence of these unmodeled states for the Q_f, Q_t, R_f, and R_t used.

As with Lashlee's thesis [8:198], the values of the noise measurement noise covariance matrices (R_f and R_t) play a role in the performance of the MMAE and MMAC algorithms. Likewise, the value of the dynamics noise strength matrices (Q_f and Q_t) also play a role in the performance of these algorithms. Proper tuning of the R and Q matrices must be accomplished for accurate

MMAE and MMAC performance. As demonstrated by Lashlee, the truth model Q_t and R_t matrices must not be so large that they mask out the difference between good and poor models. In the same manner, the filter model Q_f and R_f matrices must be properly tuned to account for the unmodeled effects as best as possible.

As noted previously, acceptable MMAE performance can only be achieved when the moving bank of elemental Kalman filters is initially centered at the true parameter point. This performance is not acceptable for a "real world" scenario where the true parameter is most likely not known. Likewise, the fact that the moving bank cannot accurately estimate the parameters is a serious problem. Based on Lashlee's conclusion [9:21] that accurate parameter estimates are strongly correlated to the ability of the algorithms to generate accurate state estimates (and control), this inability of the algorithm to generate accurate parameter estimates is very likely the reason accurate state estimation and control is not achievable when the bank is initialized at an incorrect parameter value.

As in Lashlee's thesis, the state weighting and control weighting matrices must be tuned for parameter used. As of this thesis effort, tuning has been performed for only two parameter vector values. One parameter is at (5,5), and the other parameter is at (7,6). The values of the state and control weighting matrix tunings were performed by Lashlee and can be

found in his thesis [8:143,180]. Adequate control performance requires tuning for all parameter values, but due to time constraints the tuning was not performed for this thesis. This may be one reason the MMAC results were unstable.

6.3 Recommendations

It is strongly recommended that research be continued using the rotating two-bay flexible space structure model used in this thesis, Lashlee's thesis, and Karnick's thesis. Then, using this same space structure, the following recommendations are made:

1) Continue the robustness analysis. However, increase the number of filter model states to incorporate the modes (modes 4 through 7) which seem to be causing the robustness problems. Ideally, a 24-state filter model (incorporating all the truth model modes) should be used to achieve a measure of the best possible adaptive performance.

2) Ensure the state weighting and control weighting matrices are tuned for each parameter.

3) Investigate the performance of the MMAE/MMAC algorithms in the presence of varying parameters. In this thesis effort, the true parameter is held constant. The follow-on thesis should examine the effects of varying parameters (slowly varying or instantaneous jumps) on the adaptive algorithms.

4) Investigate what happens to the Kalman filter gains as the covariance matrices are increased for the MMAE and MMAC studies. In this thesis, the R_f and Q matrices are increased for MMAE, but the R_f and Q_t matrices are decreased for MMAC. The actual effect on the Kalman filter gains of increasing and decreasing the covariance matrices should be investigated in detail. Likewise, the software allowing for true Q_f and R_f tuning should be used for the MMAE study.

5) Investigate the reason the pseudo-sinusoidal error state estimation plots are generated for the single artificially informed non-adaptive filter as shown by Figure 23. This effect is likely due to the unmodeled states.

6) Ensure the parameter space is properly discretized. Lashlee found [8:115-117] that parameter identifiability is enhanced when the parameter space is properly discretized. The description for how this is done is found in Section 4.4.4 of Lashlee's thesis [8:81-83].

7) As another method of performing the robustness analysis, allow the last 18 columns of the truth model measurement matrix, H_t, to be scaled by a constant from zero to one. This scaling will allow for research investigation into the effects of the unmodeled states as their presence is "weighted" by the scale

factor. A scale factor of zero would indicate there are no unmodeled states, and a scale factor of one would indicate full presence of the unmodeled effects. This study could potentially provide a great deal of insight into the effects of the unmodeled effects on the adaptive algorithm. As this is done, investigate the relative magnitudes of the mode contributions by the modes included in the reduced order model versus those excluded from the reduced order model.

8) Rather than examine the estimation and control of the modal states (the rigid body and first two out of 11 bending modes) themselves, examine the estimation and control of the positions and velocities located at the hub, node 1, and node 2 of Figure 8. This would be done by using $H_t\underline{x}_t - H\hat{\underline{x}}$ rather than $Tx_t - \hat{\underline{x}}$ for the estimation error.

9) Use a full R_f matrix for a better portrayal of the unmodeled effects, rather than just a diagonal R_f matrix.

10) Further investigation into MMAC is also <u>strongly</u> recommended. The study performed in this thesis was unable to produce stable results. Stable control is the whole reason for using the adaptive algorithm.

11) Since no fixed-bank MMAE or MMAC implementation was used for this research, investigate the fixed-bank algorithm to

determine whether or not the fixed-bank MMAE or MMAC can work more robustly in the face of unmodeled effects than a moving-bank algorithm.

Appendix A: LQG Controller Development [4; 8]

The following stochastic system is assumed [1:20-22; 2:33-35]:

$$\dot{\underline{\underline{x}}}(t) = F\underline{\underline{x}}(t) + B\underline{\underline{u}}(t) + G\underline{\underline{w}}(t) \tag{84}$$

where

$$E[\underline{\underline{w}}(t)] = \underline{0} \tag{85}$$

$$E[\underline{\underline{w}}(t)\underline{\underline{w}}^T(t+\tau)] = Q\,\delta(\tau) \tag{86}$$

and the minimized quadratic cost function is:

$$J = E\{ \int_0^\infty \tfrac{1}{2}[\underline{\underline{x}}^T(t)W_x\underline{\underline{x}}(t) + \underline{\underline{u}}^T(t)W_u\underline{\underline{u}}(t)]\underline{dt} \} \tag{87}$$

where the weighting matrices, W_x and W_u, are chosen iteratively to provide a controller with acceptable performance characteristics. With the assumption of full-state access, the object is to obtain the optimal constant-gain discrete linear feedback control law given by:

$$\underline{u}(t_i) = -G_c^*\underline{x}(t_i) \tag{88}$$

where J is minimized by the constant gain, G_c^*, which is given by [15:68,122]:

$$G_c^* = [U + B_d^T K_c B_d]^{-1}[B_d^T K_c \Phi + S^T] \tag{89}$$

where K_c is found by the algebraic Riccati equation:

$$K_c = X + \Phi^T K_c \Phi - [B_d^T K_c \Phi + S^T]^T G_c^* \tag{90}$$

and

$$X = \int_{t_i}^{t_{i+1}} \Phi^T(\tau,t_i) W_x \Phi(\tau,t_i) d\tau \tag{91}$$

$$U = \int_{t_i}^{t_{i+1}} [\bar{B}^T(\tau,t_i) W_x \bar{B}(\tau,t_i) + W_u] d\tau \tag{92}$$

$$S = \int_{t_i}^{t_{i+1}} \Phi^T(\tau,t_i) W_x \bar{B}(\tau,t_i) d\tau \tag{93}$$

$$\bar{B}(t,t_i) = \int_{t_i}^{t_{i+1}} \Phi(\tau,t_i) B d\tau \tag{94}$$

$$B_d = \bar{B}(t_{i+1},t_i) \tag{95}$$

The state transition matrix from t_1 to t_2 is $\Phi(t_2,t_1)$, and Equations (89) to (94) use $\Phi = \Phi(t_{i+1},t_i)$. If no driving noise, $\underline{w}(t)$, is present, then Equation (88) is the solution for the deterministic LQ optimal control problem. By use of the certainty equivalence property [15:17], the full-state access of Equation (88), $\underline{x}(t_i)$, is replaced by the state estimate $\hat{\underline{x}}(t_i^+)$. The use of the certainty equivalence property is best described via:

> Under the LQG assumptions, the design of the optimal stochastic controller can be completely separated into the design of the appropriate Kalman filter cascaded with the optimal feedback gain matrix of the corresponding deterministic optimal control problem. [15:17]

Appendix B: Rotating Two-Bay Truss System Matrices

[4:100-107;8:205-210]

This appendix shows the truth model mass and stiffness matrices, and the 6-state filter model system matrices for the two-bay rotating truss. The order reduction is as performed in Section 3.5. The matrices are given in the modal formulation, and the filter system is composed of 6 states with 6 measurements and 3 control inputs. There are 3 position and 3 velocity states. The states are the modal positions and velocities in this order: state 1 is the rigid body position, state 2 is the first bending mode position, state 3 is the second bending mode position, state 4 is the rigid body velocity, state 5 is the first bending mode velocity, and state 6 is the second bending mode velocity. All positions are in inches and velocities are in inches per second.

Stiffness Matrix

The stiffness matrix is a 12-by-12 matrix generated by Karnick [4], used by Lashlee [8], and used by this thesis effort in Equations (48) - (50) and (57) - (60). The matrix is [8:206]:

Row 1:	1188.0	196.6	0.0	0.0	-642.4	0.0
	-546.1	-196.6	0.0	0.0	0.0	0.0
Row 2:	196.6	626.3	0.0	-555.6	0.0	0.0
	-196.6	-70.8	0.0	0.0	0.0	0.0
Row 3:	0.0	0.0	1188.0	-196.6	-546.1	196.6
	-642.4	0.0	0.0	0.0	0.0	0.0

Row 4:	0.0	-555.6	-196.6	623.6	196.6	-70.8
	0.0	0.0	0.0	0.0	0.0	0.0
Row 5:	-642.4	0.0	-541.6	196.6	4019.0	66.9
	0.0	0.0	-2099.0	0.0	-732.0	-263.5
Row 6:	0.0	0.0	196.6	-70.8	66.9	721.2
	0.0	-555.6	0.0	0.0	-263.5	-94.9
Row 7:	-546.1	-196.6	-642.4	0.0	0.0	0.0
	4019.0	-66.9	-732.0	263.5	-2099.0	0.0
Row 8:	-196.6	-70.8	0.0	0.0	0.0	-555.6
	-66.9	721.2	263.5	-94.9	0.0	0.0
Row 9:	0.0	0.0	0.0	0.0	-2099.0	0.0
	-732.0	263.5	86180.0	47880.0	0.0	0.0
Row 10:	0.0	0.0	0.0	0.0	0.0	0.0
	263.5	-94.9	47880.0	139000.0	0.0	-111100.0
Row 11:	0.0	0.0	0.0	0.0	-732.0	-263.5
	-2099.0	0.0	0.0	0.0	86180.0	-47880.0
Row 12:	0.0	0.0	0.0	0.0	-263.5	-94.9
	0.0	0.0	0.0	-111100.0	-47880.0	139000.0

Mass Matrix

When compared to the structural mass, the non-structural mass is large (refer to Section 3.4). Therefore, the first 8 diagonal elements of the mass matrix are essentially the values of the non-structural mass.

Row 1:	1.294000000	-0.000002395	0.000000000	0.000000000
	0.000006927	0.000000000	0.000006652	0.000002395
	0.000000000	0.000000000	0.000000000	0.000000000
Row 2:	-0.000002395	1.294000000	0.000000000	0.000000776
	0.000000000	0.000000000	0.000002395	0.000000862
	0.000000000	0.000000000	0.000000000	0.000000000
Row 3:	0.000000000	0.000000000	1.294000000	0.000002395
	0.000006652	-0.000002395	0.000006927	0.000000000
	0.000000000	0.000000000	0.000000000	0.000000000

Row 4:	0.000000000	0.000000776	0.000002395	1.294000000
	-0.000002395	0.000000862	0.000000000	0.000000000
	0.000000000	0.000000000	0.000000000	0.000000000
Row 5:	0.000006927	0.000000000	0.000006652	-0.000002395
	1.294000000	-0.000000815	0.000000000	0.000000000
	0.000022630	0.000000000	0.000008916	0.000003120
Row 6:	0.000000000	0.000000000	-0.000002395	0.000000862
	-0.000000815	1.294000000	0.000000000	0.000000776
	0.000000000	0.000000000	0.000003210	0.000001156
Row 7:	0.000006652	0.000002395	0.000006927	0.000000000
	0.000000000	0.000000000	1.294000000	0.000000815
	0.000008916	-0.000003210	0.000022630	0.000000000
Row 8:	0.000002395	0.000000862	0.000000000	0.000000000
	0.000000000	0.000000776	0.000000815	1.294000000
	-0.000003210	0.000001156	0.000000000	0.000000000
Row 9:	0.000000000	0.000000000	0.000000000	0.000000000
	0.000022630	0.000000000	0.000008916	-0.000003210
	0.000881700	-0.000064020	0.000000000	0.000000000
Row 10:	0.000000000	0.000000000	0.000000000	0.000000000
	0.000000000	0.000000000	-0.000003210	0.000001156
	-0.000064020	0.000834300	0.000000000	0.000155300
Row 11:	0.000000000	0.000000000	0.000000000	0.000000000
	0.000008916	0.000003210	0.000022630	0.000000000
	0.000000000	0.000000000	0.000881700	0.000064020
Row 12:	0.000000000	0.000000000	0.000000000	0.000000000
	0.000003210	0.000001156	0.000000000	0.000000000
	0.000000000	0.000155300	0.000064020	0.000834300

Filter Model F Matrix (defined by Equation (48))

$$\begin{bmatrix} 0.00000 & 0.00000 & 0.00000 & 1.00000 & 0.00000 & 0.00000 \\ 0.00000 & 0.00000 & 0.00000 & 0.00000 & 1.00000 & 0.00000 \\ 0.00000 & 0.00000 & 0.00000 & 0.00000 & 0.00000 & 1.00000 \\ 0.00000 & 0.00000 & 0.00000 & 0.00000 & 0.00000 & 0.00000 \\ 0.00000 & -79.18000 & 0.00000 & 0.00000 & -0.08898 & 0.00000 \\ 0.00000 & 0.00000 & -508.50000 & 0.00000 & 0.00000 & -0.22550 \end{bmatrix}$$

The rigid body angular position and velocity correspond to rows 1 and 4 respectively. The position of the first and second bending modes correspond to rows 2 and 3 respectively, and the

velocity of the bending modes corresponds to rows 5 and 6 respectively. The truth model F matrix is defined in Appendix C.

B Matrix

$$\begin{bmatrix} 0.00000 & 0.00000 & 0.00000 \\ 0.00000 & 0.00000 & 0.00000 \\ 0.00000 & 0.00000 & 0.00000 \\ -0.47730 & -0.27110 & 1.00000 \\ 0.25890 & -0.47360 & 0.00000 \\ 0.09488 & 0.15900 & 0.00000 \end{bmatrix}$$

Inputs from the actuators located on the truss are represented by the first two columns. The position (rigid body, first bending, and second bending) modes are represented by the first three rows, and the velocity (rigid body, first bending, and second bending) modes are represented by the last three rows, respectively. The third column corresponds to an actuator located on the hub. The first two columns were originally designed in the unreduced physical coordinate system as

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^T$$

and transformed into modal coordinates. Adding the angular input actuator is done by augmenting a third column with a 1 entered in the row corresponding to the angular velocity state (the rest of the column is all zeros). The B matrix has 24 rows for the truth model and 6 rows for the filter model.

H Matrix

$$\begin{bmatrix} 0.60660 & -0.32390 & -0.12280 & 0.00000 & 0.00000 & 0.00000 \\ 0.34440 & 0.59250 & -0.20570 & 0.00000 & 0.00000 & 0.00000 \\ 0.00000 & 0.00000 & 0.00000 & 0.60660 & -0.32390 & -0.12280 \\ 0.00000 & 0.00000 & 0.00000 & 0.34440 & 0.59250 & -0.20570 \\ 1.00000 & 0.00000 & 0.00000 & 0.00000 & 0.00000 & 0.00000 \\ 0.00000 & 0.00000 & 0.00000 & 1.00000 & 0.00000 & 0.00000 \end{bmatrix}$$

The position sensor measurements located at nodes 1 and 2 are represented by the first and second rows, and the velocity sensor measurements located at nodes 1 and 2 are represented by the third and fourth rows. The angular position and velocity measurements of the hub are represented by rows 5 and 6. The H matrix is determined by first calculating the position and velocity portion in physical coordinates as

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

and transformed to modal coordinates. After this transformation, rows 5 and 6 are augmented, representing the angular position and velocity measurements. Note that the upper right partition and the bottom right partition are the same because the position and velocity sensors are co-located. The H matrix for the filter model is 6-by-6, and the H matrix for the truth model has 24 columns.

D_r Matrix

$$\begin{bmatrix} -0.000425300 & 0.000063960 & 0.000000000 & 0.000000000 \\ 0.000063960 & -0.000358300 & 0.000000000 & 0.000000000 \\ 0.000000000 & 0.000000000 & 0.000000000 & 0.000000000 \\ 0.000000000 & 0.000000000 & 0.000000000 & 0.000000000 \\ 0.000000000 & 0.000000000 & 0.000000000 & 0.000000000 \\ 0.000000000 & 0.000000000 & 0.000000000 & 0.000000000 \end{bmatrix}$$

The method of singular perturbations is used to calculate the D_r matrix. The D_r matrix is used in the 6-state filter model, but it is not needed for the 24-state truth model.

Appendix C: Development of the 24-State F Matrix

The software used in this thesis is based on the same software used in previous theses [1; 2; 4; 5; 8; 9; 16]. However, this software was originally set up only for a 6-state truth model. It is necessary to modify the development of the F matrix to accommodate the 24-state truth model. The software now uses both a 6-state and a 24-state F matrices.

The 24-state F matrix had to be developed so that the first 6 rows and columns correspond to the 6-state filter model F matrix. This means that, rather than having 12 position states and then 12 velocity states represented by the 24 rows, the first 6 rows of the truth model F matrix represent the three position states and three velocity states of the rigid body and first two bending modes, in that order. The next nine rows following these first six rows represent position states of the next nine bending modes and the last nine rows represent the corresponding velocity states. This modified truth model F matrix is required because the measurement, propagation, and update equations of the thesis software are set up for the 6-state F matrix, and the 24-state F matrix must match the 6-state form. For this reason, the 24-state F matrix when developed as shown by Equation (60) is not usable. Rather, the 24-state F matrix needs to be transformed (using a transformation matrix T) into a form usable by the software.

The 24-state F matrix is developed by first finding the 24-state F matrix by using Equation (60). Then a transformation matrix, T, is found which will transform the 24-state F matrix into:

$$F = \left[\begin{array}{c|cc} F_{6\text{-state}} & \underline{0}_{6\text{-by }18} & \\ \hline \underline{0}_{18\text{-by-}6} & I_{9\text{-by-}9} & \underline{0}_{9\text{-by-}9} \\ & [-\omega_i^2]_{9\text{-by-}9} & [-2\ \zeta_i\ \omega_i] \end{array}\right] \qquad (96)$$

where $F_{6\text{-state}}$ is the exact 6-state F matrix used by the filter model, and the other 9 modes are incorporated in the bottom right partition of Equation (96). This F matrix is needed because the software assumes the states x_1, x_2, and x_3 are the positions of the first three modes and the states x_4, x_5, and x_6 are the velocities of the first three modes. If the 24-state F matrix is developed as given by Equation (60), then the states x_1 through x_{12} are positions and the states x_{13} through x_{24} are velocities. The transformation matrix used for this transformation is:

$$T = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix} \quad (97)$$

The truth model F, B, and H matrices need to be transformed into the form usable by the software. These transformed matrices are denoted F_A, B_A, and H_A (A stands for altered). The transformation equations are used:

$$F_A = T*F*T^{-1} \quad (98)$$

$$B_A = T*B \quad (99)$$

$$H_A = H*T^{-1} \quad (100)$$

Bibliography

1. Filios, Capt Paul G. Moving Bank Multiple Model Adaptive Algorithms Applied to Flexible Spacecraft Control. MS thesis, AFIT/GE/ENG/85D-14. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, December 1985 (AD-A164016).

2. Hentz, Lt Karl P. Feasibility Analysis of Moving Bank Multiple Model Adaptive Estimation and Control Algorithms. MS thesis, AFIT/GE/ENG/84D-32. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, December 1984 (AD-A152015).

3. Kailath, Thomas. Linear Systems. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1980.

4. Karnick, Lt Drew A. Moving Bank Multiple Model Adaptive Estimation Applied to Flexible Spacestructure Control. MS thesis, AFIT/GE/ENG/86D-41. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, December 1986 (AD-A178870).

5. Karnick, Lt Drew A., and Peter S. Maybeck. "Moving Bank Multiple Model Adaptive Estimation Applied to Flexible Spacestructure Control," Proceedings of the 26th IEEE Conference on Decision and Control, Los Angeles, California, pp. 1249-1257 (December 1987).

6. Kokotovic, P. V., R. E. O'Malley, Jr., and P. Sannuti. "Singular Perturbations and Order Reduction in Control Theory--An Overview," Automatica, 12: 123-132 (1976).

7. Kokotovic, P. V., and R. A. Yackel. "Singular Perturbation of Linear Regulators: Basic Theorems," IEEE Transactions on Automatic Control, Vol. AC-17, No. 1, pp. 29-37 (1972).

8. Lashlee, Capt Robert W., Jr. Moving-Bank Multiple Model Adaptive Estimation Applied to Flexible Spacestructure Control. MS thesis, AFIT/GE/ENG/87D-36. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, December 1987 (AD-A190761).

9. Lashlee, Capt Robert W., Jr., and Peter S. Maybeck. "Spacestructure Control Using Moving Bank Multiple Model Adaptive Estimation", Proceedings of the 27th IEEE Conference on Decision and Control, Austin, Texas. (December 1988).

10. Loving, Capt Phyllis A. Bayesian vs MAP Multiple Model Adaptive Estimation for Field of View Expansion in Tracking Airborne Targets. MS thesis, AFIT/GE/ENG/85M-1. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, December 1985.

11. Lynch, P. J., and Siva S. Banda. "Active Control for Vibration Damping," Damping: 1986 Proceedings, Technical Report. Flight Dynamics Laboratory AFWAL/FIGC, Wright-Patterson AFB, OH, 1986 (AFWAL-TR-86-3509).

12. Maybeck, Peter S. Unpublished class handout distributed in EENG 765, Stochastic Estimation and Control I, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, January 1988.

13. Maybeck, Peter S. Stochastic Models, Estimation and Control Volume 1. New York: Academic Press, 1979.

14. Maybeck, Peter S. Stochastic Models, Estimation and Control Volume 2. New York: Academic Press, 1982.

15. Maybeck, Peter S. Stochastic Models, Estimation and Control Volume 3. New York: Academic Press, 1982.

16. Maybeck, Peter S., and Karl P. Hentz. "Investigation of Moving-Bank Multiple Model Adaptive Algorithms," AIAA Journal of Guidance, Control and Dynamics, Vol. 10, No. 1, pp. 90-96 (Jan.- Feb. 1987).

17. Meirovitch, Leonard. Analytical Methods in Vibrations. New York: The Macmillan Company, 1967.

18. Reid, J. Gary. Linear System Fundamentals. New York: McGraw-Hill Book Company, 1983.

19. Venkayya, V. B., and V. A. Tischler. "Frequency Control and Its Effects on the Dynamic Response of Flexible Structures," AIAA Journal, 23(11): pp. 1768-1774 (November 1985).

20. Venkayya, V. B., and V. A. Tischler. 'Analyze'-- Analysis of Aerospace Structures with Membrane Elements. Technical Report. Analysis and Optimization Branch, Structural Mechanics Division, Air Force Flight Dynamics Laboratory, Air Force Wright Aeronautical Laboratories, Air Force Systems Command, Wright-Patterson AFB, OH, December 1978 (AFFDL-TR-78-170).

Vita

Captain Daniel Francis Van Der Werken, Jr. was born [redacted] to D[redacted] and S[redacted] Van Der Werken. He graduated from high school in Salem, Virginia [redacted] and attended Virginia Tech. While at Virginia Tech, he enrolled in the Air Force's College Senior Engineer Program (CSEP) in 1983 and then graduated in 1984 with a BS in Electrical Engineering. After receiving his Air Force commission in September 1984, he was assigned to the 4950th Test Wing, Electronic Countermeasures Branch. He worked as a flight test engineer until May 1987 when he was assigned to the Air Force Institute of Technology. He has a wife, [redacted] and a daughter, [redacted] They are expecting another child [redacted]

[redacted]

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) AFIT/GE/ENG/88D-59	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION School of Engineering 6b. OFFICE SYMBOL (If applicable) AFIT/ENG	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State, and ZIP Code) Air Force Institute of Technology Wright-Patterson AFB, Ohio 45433	7b. ADDRESS (City, State, and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification) A Robustness Analysis of Moving-Bank Multiple Model Adaptive Estimation and Control of A Large Flexible Space Structure.

12. PERSONAL AUTHOR(S)
Daniel F. Van Der Werken, Jr., B.S.E.E., Captain, USAF

13a. TYPE OF REPORT	13b. TIME COVERED FROM ____ TO ____	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
		1988 December	236

16. SUPPLEMENTARY NOTATION

17. COSATI CODES		
FIELD	GROUP	SUB-GROUP
09	05	

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Robustness Analysis, Adaptive Control, Adaptive Estimation, Multiple Model Adaptive Estimation, Multiple Model Adaptive Control; Adaptive Filters, Thesis.

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

TITLE: A Robustness Analysis of Moving-Bank Multiple Model Adaptive Estimation and Control Applied to A Large Flexible Space Structure.

THESIS CHAIRMAN: Dr. Peter S. Maybec , Professor of Electrical Engineering

Approved for release in accordance with AFR 190-1
10 Jan 89

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL Dr. Peter S. Maybeck, Prof. of Elec. Eng.	22b. TELEPHONE (Include Area Code) 513-255-2057
	22c. OFFICE SYMBOL AFIT/ENG

DD Form 1473, JUN 86 — Previous editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

Abstract

This thesis study is a robustness analysis of a moving-bank multiple model adaptive estimation and control algorithm. The truth model consists of 24 states, and the filter model consists of 6 states. The object is to determine whether or not the state mismatch between the filter and truth models seriously degrades or totally confounds the adaptation process.

A rotating two-bay flexible space structure, approximating a space structure that has a hub with extending appendages, is the system model. The mass of the hub is greater than the mass of the appendage. Finite element analysis is used to develop the mathematical model. The system is developed in physical coordinates, transformed to modal coordinates, and the method of singular perturbations is used to provide a reduced filter model. The two dimensional parameter space consists of variations of the mass and stiffness matrices for the two-bay truss.

Results indicate stable and accurate state estimation when the bank is initially centered on the true parameter. No stable control was achieved. Accurate parameter estimates are not achieved. When the bank is initially centered at an incorrect parameter, stable and accurate state estimation and control is not demonstrated. These results indicate a total confounding of the adaptation process due to the unmodeled states, since accurate estimation and control can be generated when there are no such unmodeled states.